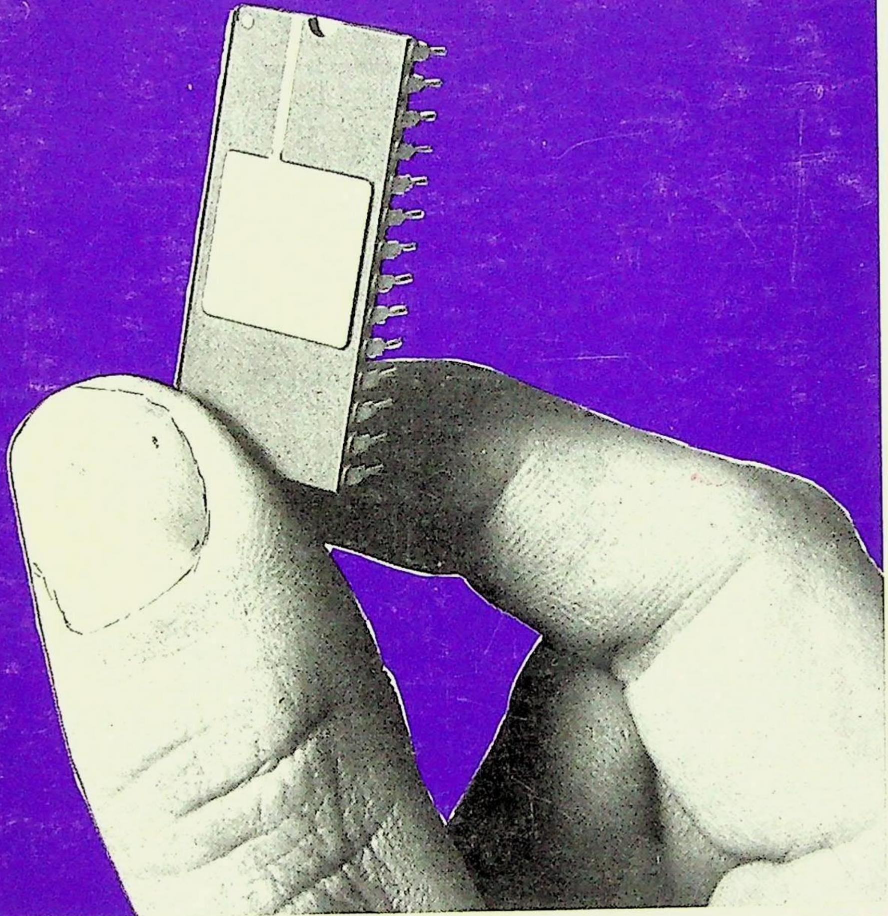
מיקרומחשבים וה- 6800
חומרה ותוכנה

אהרון הראל, B.Sc.T.E



עורך יצחק עמיהוד M.Sc

ספרי לימוד והכשרה במדעי המחשב
הוצאת הוד- עמי

מיקרומחשבים
וה- 6800
חומרה ותוכנה

מיקרומחשבים וה- 6800
חומרה ותוכנה

אהרון הראל, B.Sc.T.E

עורך
יצחק עמיהוד, M.Sc

ספרי לימוד והכשרה במדעי המחשב
הוצאת הוד- עמי

Microcomputers and the 6800
Hardware and Software

by Aaron Harel, B.Sc.T.E
edited by Itzhak Amihud, M.Sc

הוצאת הוד-עמי
לספרי מחשבים
ת.ד. 560 רמת-גן 52105

Hod–Ami
P.O.Box 560, Ramat–Gan

הודפס בישראל
חשון תשמ"ד, נובמבר 1983
Israel, November 1983

סדר במערכת מילולית מתוצרת
אבמת מחשבים בע"מ, נחלת יצחק 58, ת"א
טל. 217957, 252418

סדר ועריכה גרפית: אקזוטיק

מוקדש באהבה ובהוקרה
לרעייתי לילי
ולילדי אילן, גל וצחי

תודתי נתונה לד"ר אייב כהן, ראש המחלקה למחשבים ומנהל מעבדת אלקטרוניקה במרכז לחינוך טכנולוגי בחולון, על הזמן והמאמץ שהשקיע בהדרכה ובהכוונה.

למר יקותיאל פקטה, מרצה בכיר בתחום החינוך הטכנולוגי במרכז לחינוך טכנולוגי בחולון ומנהלו לשעבר של ביה"ס "בזק" של משרד התקשורת בירושלים, על התדרוך בדרך הגשה נכונה של חומר לתלמיד ועל המאמצים הרבים שהשקיע בקריאה, הערות והארות דידקטיות של מחנך דגול.

לעמיתי יעקב אבני, על קריאת חלק מכתב היד והארותיו העניניות.

הקדמה

המחשב מצעיד את הידע המצטבר בעולם ואת שירותי עיבוד הנתונים בתנופה שהאנושות לא ידעה כמוה בתולדותיה. אנו נמצאים בעיצומה של "מהפכת מחשבים" הסוחפת כל תחום מחיינו, החל בציוד מתוחכם ברפואה, תעשיה, צבא, חקר החלל, עיבודי תמליל וכלה במשחקים ממוחשבים, ועוד. מחשב ניתן לניצול בשטחים רבים ושונים כיד הדמיון הטובה עלינו. פיתוח טכנולוגיות חדישות איפשר להביא את המחשב להישג ידו של כל החפץ בכך. במחיר מכונית אפשר לרכוש כבר היום מערכת מחשב רב-עוצמה או כ-20 מיקרומחשבים ביתיים בעלי אפשרויות ביצוע רבות.

פיתוח הנושא נמשך ברחבי העולם בקצב גובר והולך וצפויים ללא ספק חידושים והפתעות לרוב! נושאי המיחשוב והכרת המחשב צריכים להיות נהירים לאדם המעוניין להתקדם עם הטכנולוגיה. בספר זה אין בכוונתנו לעסוק בכל תחום המיחשוב אלא לטפל בנושא המיקרומחשבים בלבד. אלו הם מחשבים זעירים אשר מותקנים במכשירים שונים ומהווים בסיס למערכות פשוטות ומורכבות כאחד.

הספר כולל סקירת נושאים תיאורטיים בחומרה ובתוכנה ושאלות לחזרה. מומלץ לערוך תרגול בתוכנה וניסויים במחשב כדי שלימוד כל קטע חומר תיאורטי ילווה בעבודה מעשית לצורך אימות, שינון ומשוב מיידי. בפרקים 1, 2 ו-3 רוכז חומר עיוני כללי בחומרה ובתוכנה כאשר הדגש הוא על המשותף לכל סוגי המיקרומחשבים. פרקים 4, 5, 6 ו-7 עוסקים בחומרה ובתוכנה של מערכת עם מיקרומעבד מסוג 6800. הפרק ה-8 הוא הפרק המסכם, בו נערכת השוואה בין שלושת סוגי המיקרומעבדים הנפוצים כיום: 6800, 8085 ו-Z-80. גם ניתנת בו הערכה לגבי מגמות הפיתוח העתידי בנושא. הביבליוגרפיה מפרטת את הספרות המתאימה להעשרת החומר הנלמד. הנספחים מציגים בין השאר נתונים טכניים שונים שנלקחו מפרסומי חברת מוטורולה וחברת אינטל ולסיום מובאים מילון מונחים אנגלי-עברי קצר. בראש כל פרק מוצג מילון קצר של המונחים החדשים בפרק לפי סדר הופעתם.

על מנת להפיק את מלוא התועלת מלימוד לפי ספר זה רצוי להכיר תחילה את הנושאים הבאים: חישובים אריתמטיים בבסיס ספירה בינרי והקסדצימלי, יסודות האלגברה הבוליאנית, מעגלים לוגיים, סוגי זכרונות, תרשימי זרימה ועקרונות תכנות.

הספר נכתב על ידי מורה לאלקטרוניקה ולמיקרומחשבים והוא מיועד למהנדסים ולטכנאים, לסטודנטים, לתלמידים, לאנשי מקצוע ולכל מי שמתעניין בתחום חשוב זה.

עריכת הספר והצגת הנושאים נעשתה על פי הדרישות של תוכנית הלימודים החדשה של משרד החינוך והתרבות לשנת תשמ"ג/ד עבור מגמות אלקטרוניקה/מחשבים. תוכנית הלימודים מכוונת לתלמידים בכיתות הגבוהות של בתי-ספר תיכוניים-מקצועיים והיא מתאימה גם לבתי-הספר להנדסאים ולטכנאים אשר בהם לומדים את הנושא. הספר כתוב במתכונת ללימוד עצמי.

תוכן העניינים

פרק 1 - מחשב מהו ?

מטרות ביצועיות לימודיות

עם סיום העיון בפרק זה הקורא יוכל:

1. לשרטט דיאגרמה מלבנית של מערכת מחשב בסיסית,
2. להסביר בקצרה תפקידו של כל מרכיב במערכת המחשב הבסיסית,
3. להסביר בקצרה מהי תכנית,
4. להסביר בקצרה מהי הוראה, ומהו מבנה הוראה,
5. להבחין בין קוד פעולה, שדה הכתובות ושדה הנתונים בהוראה,
6. להבחין בין מחשב גדול, מינימחשב, מיקרומחשב וחישובית,
7. להסביר מונחים חדשים.

מונחים בפרק זה:

יחידת עיבוד מרכזית, יע"מ	CENTRAL PROCESSING UNIT; CPU
זכרון	MEMORY
ציוד קלט	INPUT
ציוד פלט	OUTPUT
תקליט, דיסק	DISK
תקליטון	DISKETTE
רמה לוגית נמוכה	LOGIC LOW
רמה לוגית גבוהה	LOGIC HIGH
סיבית	BIT
מילה	BYTE, WORD
הוראה	INSTRUCTION
קוד פעולה	OPERATION CODE; OP CODE
כתובת	ADDRESS

1.1 מבוא

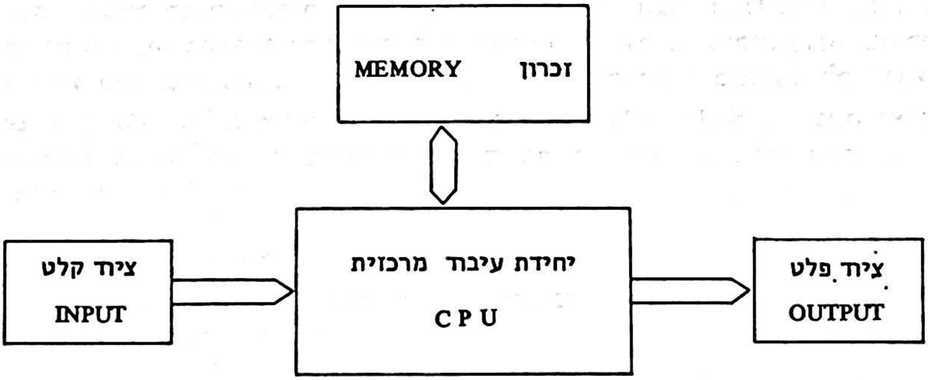
למחשב שלוש תכונות יסוד:

א. אגירת מידע בכמות גדולה,

ב. יכולת לעבד מידע בגמישות רבה,

ג. מהירות פעולה גבוהה.

1.2 מבנה מערכת מחשב בסיסית



איור 1.1 – תרשים מלבנים של מערכת מחשב

מערכת מחשב נראית במבט ראשון מורכבת ומסובכת. המערכת בנויה מרכיבים רבים אולם היא כוללת ארבעה אלמנטים בסיסיים בלבד כפי שמתואר באיור 1.1 .

א. **יחידת עיבוד מרכזית** (CENTRAL PROCESSING UNIT) היא "הלב" של המערכת. היא מבצעת את עיבוד הנתונים, מבקרת ומתזמנת את פעולת כל יחידות המערכת.

ב. **זכרון** (MEMORY) הינו מאגר מידע מסוגים שונים. מאוחסנים בו נתונים המיועדים לעיבוד, הוראות ביצוע, תוצאות העיבוד ועוד.

ג. **ציוד קלט** (INPUT) מיועד ליצור גישה אל המערכת ותפקידו להזינה בנתונים. עליו לתרגם את הקלט המקורי הכתוב בשפת המתכנת לשפת המחשב. קיימים סוגים רבים של ציוד קלט בהתאם לשיטת הכנסת המידע, כגון לוח מקשים, סרט מגנטי, סרט מנוקב, תקליט (DISK), תקליטון (DISKETTE) ועוד.

ד. **ציוד פלט** (OUTPUT) מקבל את התוצאות הסופיות של תהליך עיבוד הנתונים. תפקידו לתרגם תפוקת המחשב לפלט אשר מתאים לציוד המבוקר. קיימים סוגים רבים של ציוד פלט בהתאם ליעוד, אך לא נעסוק בכך כאן.

1.3 על מה חושב המחשב?

המחשב לא חושב כלל. יותר מכך - הוא אינו מסוגל לחשוב אלא רק לחשב. מסקנה מיידית: אין לחשוש מתחרות בין אדם למחשב. האדם הוא החושב והוא היודע כיצד לפתור בעיה. המחשב הוא המבצע את החישובים, הבדיקות, הקלט והפלט. יתרונו מתבטא במהירות הביצוע הגבוהה פי מליוני מונים מזו של האדם.

1.4 מהי שפת המחשב?

שפת המחשב פשוטה ביותר וכוללת שני סימנים בלבד: 0 ו-1. אפשר להציג אותה בשתי רמות מתחים בלבד: הנמוכה (LOW) והגבוהה (HIGH) או רמת לוגית 0 ורמה לוגית 1 בהתאמה. לנו נוח יותר לחשוב על סיבית (BIT) בינרית 0 או 1 בהתאמה. צירופים שונים של שתי הרמות האלה מרכיבות מילה (BYTE או WORD). מילים מספר מרכיבות הוראה (INSTRUCTION). מעתה ואילך נשתמש במונח סיבית בינריות במקום רמות מתחים לוגיות, לשם נוחות וכמקובל.

1.5 מהי תוכנית?

תוכנית היא צירוף הוראות מוגדר. מטרתה להדריך את המחשב בפתרון בעיה נתונה, צעד אחר צעד.

1.6 מהי הוראה?

הוראה היא הנחייה לביצוע שלב מסויים בתהליך עיבוד הנתונים.

1.7 מבנה הוראה

כל הוראה צריכה לכלול מילה אחת לפחות. המילה הזו מהווה קוד ההפעלה (OPERATION CODE או בקיצור OP. CODE). אשר מורה למחשב מה לעשות, איך לעשות וכיצד לפנות לנתונים. בהוראות בנות שתים ושלוש מילים המילה השניה ו/או השלישית היא נתון או כתובת של נתון.

הוראה בת מילה אחת:			קוד פעולה OP. CODE
הוראה בת שתי מילים:		מילה 2	קוד פעולה
הוראה בת שלוש מילים:	מילה 3	מילה 2	קוד פעולה

1.8 היכן מסתתרות ההוראות?

ציוד הקלט מעביר את ההוראות המוצגות בשפת המכונה אל הזכרון. כל מילה נכנסת לתא נפרד שכתובתו ידועה.

1.9 מהי כתובת?

בזכרון ישנם תאים רבים מאוד.

כתובת (ADDRESS) היא אמצעי למיעון תא מסויים מבין כל התאים. כל כתובת, כמו הוראה או נתונים, מורכבת גם כן מצירוף מסויים של סיביות בינריות.



1.10 כיצד מבחין מחשב בין הוראות, כתובות ונתונים?

במחשב מוצגים ההוראות, כתובות ונתונים כסדרות של מספרים בינריים, ולכן דרוש אמצעי מוסכם כדי להבחין ביניהם. אמצעי זה הוא סדר ההוראות. להבהרת הדבר נניח הוראה הבנויה משלוש מילים אשר מאוחסנות בזכרון בתאים נפרדים כשכתובתם מוגדרת חד-משמעית בסדר עולה. קוד הפעולה קובע בין השאר מה מבטאות המילים הבאות בהוראה: כתובת או נתון.

בהמשך נשתמש במונע "מידע" (INFORMATION) כדי לבטא תכולה של תא זכרון אשר יכול להכיל נתון כלשהו, הוראה, כתובת ועוד.



1.11 מה בין מחשב גדול למינימחשב ולמיקרומחשב?

בין מינימחשב ומיקרומחשב אין הבדלים עקרוניים, אך קיימים מספר גורמים אשר מקובלים לקביעת תחומי ההפרדה:

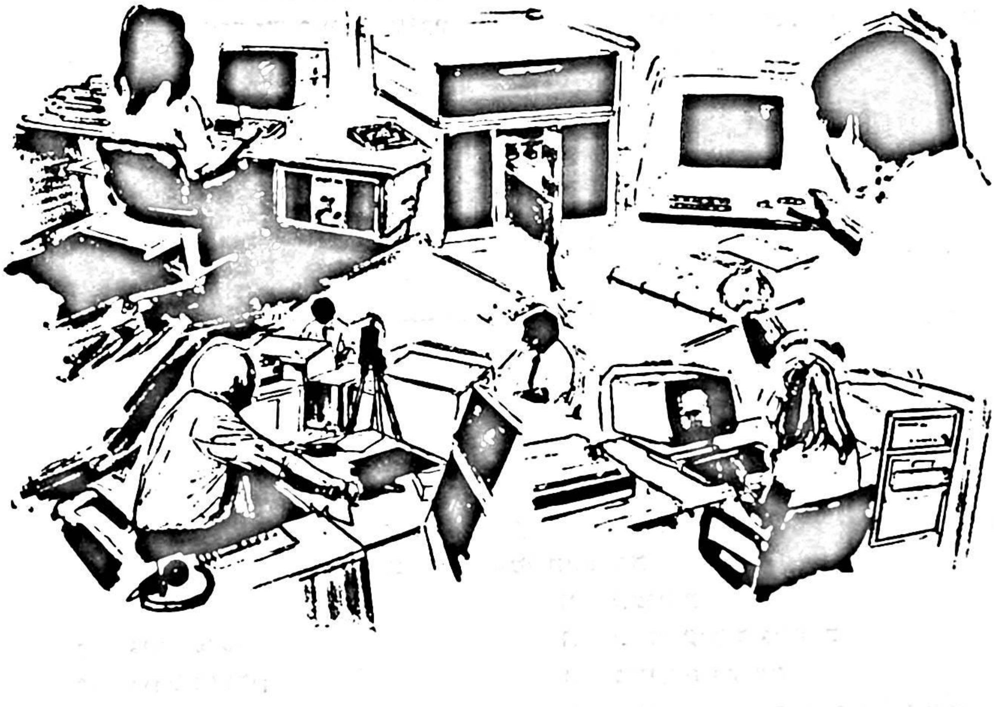
א. מחיר
ב. גודל פיסי
ג. גודל הזכרון
ד. מספר הסיביות במילה
ה. שפות תכנות
ו. כושר עיבוד נתונים
ז. מהירות פעולה
ח. אפשרות חיבור ציוד קלט/פלט

כמה מההבדלים האלה הולכים ואובדים במהירות עקב ההתפתחות הטכנולוגית.

1.12 מה בין מחשב לחישובית?

מחשב מבצע עיבוד נתונים בהתאם להוראות התוכנית שאוחסנה קודם לכן בזכרון. לחישובית לעומת זאת אין כלל זכרון המיועד לאגירת הוראות וממילא אין לה כל תוכנית. האדם המפעיל מתקתק הוראה והחישובית מבצעת מיד ללא תוכנית קבועה מראש.

קיימות בשוק חישוביות של יצרנים שונים אשר יכולות לאגור מספר פקודות ולחזור עליהן בהתאם להוראות המשתמש. חישוביות אלו אינן רק "מכונות חישוב" במובן המקובל אלא נמצאות בתחום המיקרומחשבים.



1.13 שאלות חזרה

1. שרטט דיאגרמת מלבנים של מערכת מחשב בסיסית והסבר תפקידו של כל מלבן.

2. במפעל מסוים מייצרים מכשיר קשר. הוחלט לתכנן מערך ממוחשב לבדיקת המוצר. על המחשב למדוד מתחים חשמליים במאה נקודות שונות במכשיר ולהתריע ע"י נורית במקרה של תקלה במכשיר, כולל הודעה מודפסת על מיקום המדידה השגויה.
   א. מהו ציוד הקלט הדרוש לביצוע המשימה?
   ב. מהו ציוד הפלט לביצוע המשימה?

3. ציין שלבים עיקריים בתוכנית לבדיקת מכשיר הקשר מהשאלה הקודמת כאשר:
   א. הבדיקה תתבצע בסדר מסוים, נקודה אחר נקודה.
   ב. הבדיקה תתבצע בבת אחת בכל מאה הנקודות.

4. המתכנת מעביר תוכנית למחשב בעזרת המקלדת. מה קורה לתוכנית?

5. הוראת מחשב עשויה לכלול מהות הפעולה, נתונים, כתובות. כיצד מבחין המחשב ביניהם?

6. נניח מילת מחשב 10011100.
   ציין רצף של רמות המתחים החשמליים במחשב עבור מילה זו.

7. מהו קוד הפעולה?

פרק 2 - יחידת עיבוד מרכזית

מטרות ביצועיות לימודיות

לאחר סיום עיון בפרק זה הקורא יוכל :

1. לשרטט תרשים מלבנים של מיקרומעבד
2. לשרטט ולהסביר מבנה פנימי של מיקרומעבד אופייני
3. למנות לפחות ארבעה סוגי אוגרים לשימוש המתכנת
4. למנות ולהסביר שלבי ביצוע הוראה במחשב
5. למנות לפחות שישה סוגי אוגרים ביע"מ הלוקחים חלק בביצוע הוראה
6. להסביר מונחים חדשים המופיעים בפרק זה.

מונחים בפרק זה:

מיקרומעבד	MICROPROCESSOR; MP
יחידה אריתמטית לוגית	ARITHMETIC LOGIC UNIT; A.L.U.
יחידת הבקרה	CONTROL UNIT; C.U.
צובר	ACCUMULATOR
אוגר	REGISTER
הזנת מתח	POWER
שעון מרכזי	CENTRAL CLOCK
קוי כתובת	ADDRESS BUS; A.B.
קוי נתונים	DATA BUS; D.B.
קוי בקרה	CONTROL BUS; C.B.
זיווד בצפיפות גבוהה	LARGE SCALE INTERGRATION; L.S.I.
אוגר זמני	TEMPORARY REGISTER
אוגר הדגלים	FLAG REGISTER
אוגר ההוראות	INSTRUCTION REGISTER; I.R.
מונה תכנית	PROGRAM COUNTER; P.C.
מצביע מחסנית	STACK POINTER; S.P.
אוגר מורה	INDEX REGISTER; IX
אחרון נכנס - ראשון יוצא	LAST IN FIRST OUT; L.I.F.O.
(שלב) הבאה	FETCHING
(שלב) ביצוע	EXECUTION
קריאה	READ
כתיבה	WRITE
מחזור המכונה	MACHINE CYCLE
מחזור השעון	CLOCK CYCLE

2.1 מבוא

יחידת עיבוד מרכזית (CENTRAL PROCESSING UNIT) או בקיצור יע"מ (C.P.U.) מהווה את "המוח" ו"מרכז העצבים" של מערכת המחשב. שני תפקידיה העיקריים הם ביצוע עיבוד נתונים ופיקוח ותזמון של המערכת. גודלה ומורכבותה תלויים בהיקף המערכת ומספר הסיביות במילת מחשב. במערכת המיקרומחשב היא בנויה ומזוודת במעגל משולב ונקראת גם מיקרומעבד (MP MICROPROCESSOR;). בפרק זה יתואר מבנה כללי של מיקרומעבד אופייני. יש לציין כי קיים מבנה משותף לכל המיקרומעבדים הנפוצים כיום.

2.2 תרשים מלבנים של מיקרומעבד

ניתן להבחין במחשב בשלושה מרכיבים עיקריים (ראה איור 2.1):



איור 2.1 – תרשים מלבנים של מיקרומעבד

א. יחידה אריתמטית לוגית (Arithmetic Logic Unit; A.L.U.)

יחידה אריתמטית לוגית מבצעת חישובים אריתמטיים כגון חיבור, חיסור, כפל וחילוק והיא מבצעת פעולות לוגיות כגון גם (AND) או (OR) וכד'.

ב. יחידת בקרה (Control Unit; C.U.)

תפקידי יחידת הבקרה העיקריים הם שניים:

1) פיקוד ומעקב אחר פעולות היע"מ וכלל מערכת המחשב על כל מרכיביה.

2) תאום ותזמון בין יחידות המערכת והציוד המבוקר החיצוני.

ג. צוברים ואוגרים (ACCUMULATORS AND REGISTERS)

הצוברים והאוגרים מאחסנים הוראות ונתונים זמניים לצורך ביצוע פעולה שוטפת ושומרים תוצאות עיבוד ביניים עד לאיחסונם מחוץ ליע"מ.

ד. קווי בקרה (Control Bus; C.B.)

קוי הבקרה הינם קוי יציאה וקוי כניסה של המעבד. לכל קו תפקיד מוגדר והוא אינו קשור לקו אחר. תפקיד קוי היציאה הוא להודיע לשאר מרכיבי המערכת על מצב שבו נמצא היע"מ. תפקידי הקוים הנכנסים להודיע ליע"מ על מצב מרכיבי המערכת.

ה. קוי הנתונים (Data Bus; D.B.)

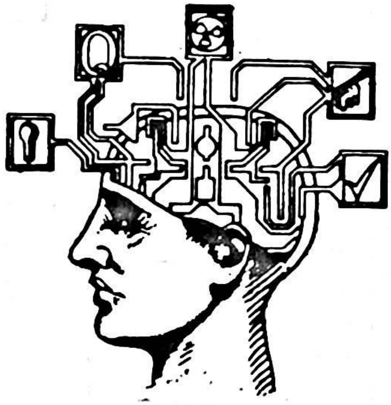
אלו הם קוים דו-כווניים דרכם עוברים נתונים אל היע"מ וממנו על פי שליטת יחידת הבקרה. צירוף מסויים של רמות לוגיות בקוים אלה ברגע נתון מהווה מידע. מספר הקוים הינו כמספר הסיביות במילת מחשב אחת.

ו. קוי הכתובת (Address Bus; A.B.)

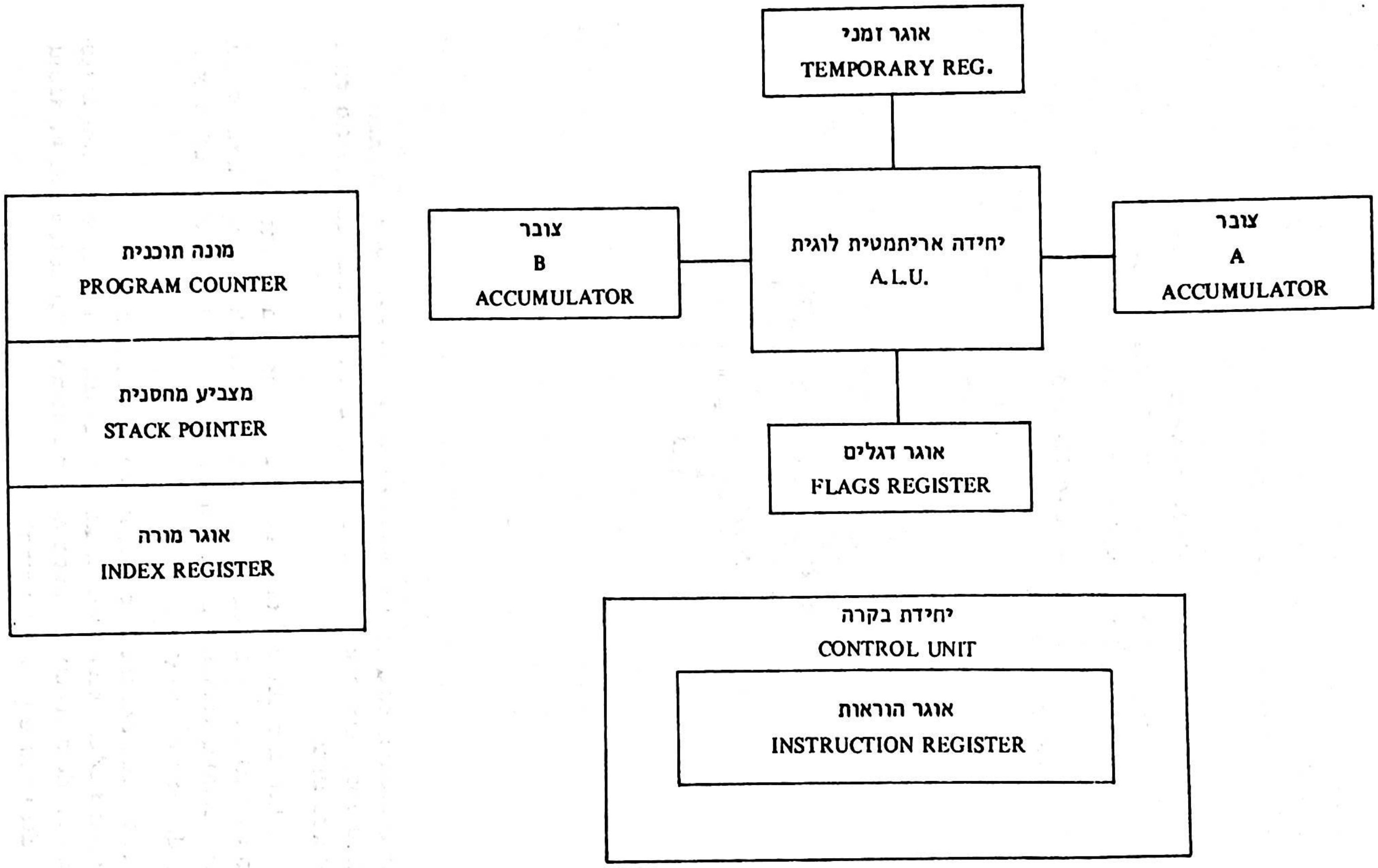
קוי הכתובת יוצאים מהיע"מ. צירוף מסויים של רמות לוגיות בקוים אלה ברגע נתון מהווה כתובת של תא בזכרון או בהתקן קלט/פלט כלשהו שאליו פונה היע"מ באותה עת. מספר הקוים הינו כמספר הסיביות במילה.

ז. מתח הזנה

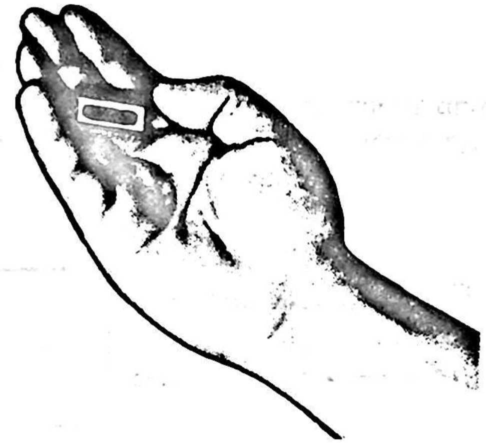
מתח הזנה הוא מתח ישר אשר דרוש להפעיל את המעגלים השונים בתוך המיקרומעבד, כשם שדרושה אספקת מתח ישר לכל מעגל אלקטרוני שהוא.



יש רק דמיון אך לא שוויון



איור 2.2 – מבנה עקרוני של מיקרומעבד



ח. שעון המערכת

דרך כניסות השעון מגיעים דפקי השעון המרכזי (CENTRAL CLOCK PULSES) אשר מאפשרים ליחידת הבקרה לתזמן את פעולת כל מרכיבי המערכת. מספר הקוים המתחברים ליע"מ מגיע כיום ל-40. במיקרומעבדים מסויימים מותקן מעגל שעון פנימי ואז הכניסות הן חיבור גביש לצורך ייצוב תדר השעון. תדירות השעון נמדדת בסדר גודל של MHz אחדים.

2.3 מבנה פנימי של מיקרומעבד אופייני

מיקרומעבד בנוי כמעגל משולב אחד, אולם הוא כולל עשרות אלפי רכיבים פעילים וסבילים. יוצא מכך שהוא מזווד בצפיפות גבוהה (Large Scale Integration ; L.S.I.) וקשה לעקוב אחר המעגל המלא והמפורט. למעשה הדבר גם איננו חיוני, מאחר וניתן להבין את עקרונות פעולת המיקרומעבד לפי תרשים המבנה הבסיסי (ראה איור 2.2)

2.4 יחידה אריתמטית לוגית

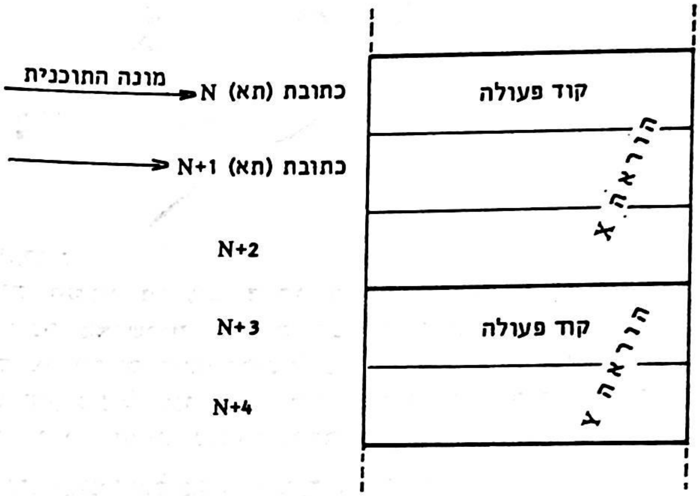
יחידה אריתמטית לוגית מבצעת כאמור פעולות אריתמטיות ולוגיות. לשם כך היא נעזרת בצוברים A ו-B (ACCumulator A, ACCumulator B) ובאוגר זמני (TEMPorary REGister) המספקים לה נתונים המיועדים לעיבוד. דרך הצוברים A ו-B היא גם מחזירה את תוצאות הפעולה (במיקרומעבדים מסויימים קיים צובר אחד בלבד). לאוגר הדגלים (FLAGS REGister) מספקת היחידה מידע על תוצאות הפעולה, כגון תוצאה חיובית או שלילית ועוד. אורכם של הצוברים והאוגר הזמני כאורך מילה אחת של מיקרומעבד שהיא לרוב שמונה סיביות.

2.5 יחידת הבקרה

יחידת הבקרה היא "מרכז העצבים" של המערכת כולה. היא מעין "המטה הכללי" המניע, המבקר והמתזמן את הפעולות של מרכיבי המערכת. אחד מתפקידיה הוא לפענח את מהות ההוראה שיש לבצע ולשם כך היא נעזרת באוגר ההוראות (Instruction Register; I.R.). הוא מאחסן את המילה הראשונה בהוראה המהווה את קוד הפעולה (O.P. CODE, Operation CODE). אורכו של אוגר ההוראה כאורכה של מילת מחשב אחת שהיא שמונה סיביות לרוב.

2.6 אוגר מונה תוכנית

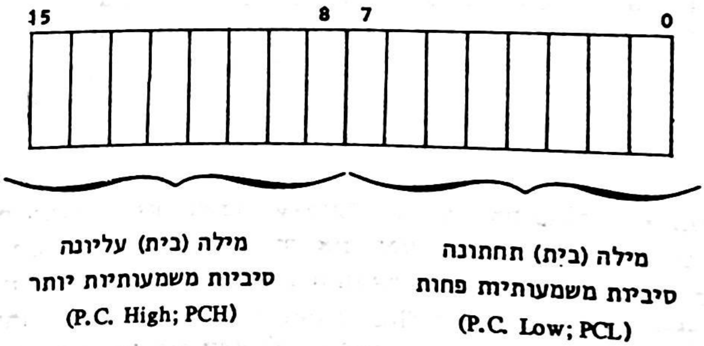
מונה התוכנית (Program Counter; P.C.) הוא אוגר שתוכנו מהווה כתובת של תא בזכרון שבו נמצאת מילת ההוראה שיש לבצע (ראה איור 2.3).



איור 2.3 - פעולת אוגר מונה התוכנית

תפקידו של מונה התוכנית לספק את כתובות התאים בזה אחר זה (כלומר, מילה אחר מילה) בקצב ביצוע התוכנית. נניח לדוגמא כי ברגע מסויים תוכנו של מונה התוכנית הוא הכתובת N. לאחר העתקת המילה שבתא N לאוגר ההוראות לצורך פענוח קוד הפעולה, תוכנו של מונה התוכנית גדל אוטומטית באחד והוא מצביע על הכתובת (התא) N+1 שבה נמצאת המילה הבאה בתור.

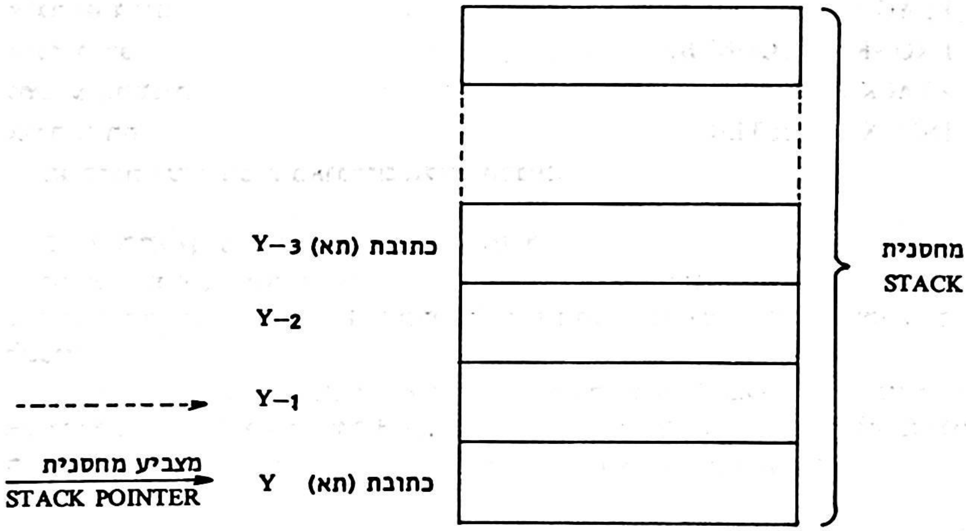
זאת היא דרך פעולתו הרגילה, אולם ניתן לאלצו להצביע על כתובת של תא כלשהו בזכרון. גורמות לכך הוראות ההסתעפות המפורטות בפרק 7. אורך הכתובת במרבית המיקרומחשבים כיום הוא 16 סיביות בינריות. מכאן שאורכו של אוגר מונה התוכנית הוא 16 סיביות שהן שתי מילים (ראה איור 2.4).



איור 2.4 - מונה התוכנית P.C.

2.7 אוגר מצביע מחסנית

מצביע מחסנית (Stack Pointer; S.P.) הוא אוגר שתוכנו הוא כתובת של תא פנוי באזור הזכרון המכונה מחסנית (Stack). אם מכניסים מידע לתא יצביע האוגר אוטומטית על כתובת התא הפנוי הבא במחסנית (ראה איור 2.5).



איור 2.5 פעולת אוגר מצביע המחסנית

סדר הכנסת המידע הינו החל מכתובת הגבוהה ביותר במחסנית (משמאל לכיוון הכתובת הנמוכה יותר (מימין). נניח שהוכנס מידע לתא Y. מיד לאחר הכנסתו יציין מצביע המחסנית אוטומטית את הכתובת Y–1. בהגיע מילה חדשה היא תאוחסן בתא Y–1. תוכן האוגר יקטן באחד ויצביע על הכתובת הפנוייה הבאה שהיא Y–2, וכן הלאה.

סדר הוצאת המידע הינו בכיוון ההפוך, כלומר החל בכתובת הנמוכה ביותר (מימין) שבה התמלאה המחסנית אל הכתובת הגבוהה יותר. בשיטת עבודה זו מידע שהוכנס אחרון ייצא ראשון (LAST IN FIRST OUT; L.I.F.O.). דוגמאות רבות לשיטת עבודה כזאת ניתן למצוא בחיינו היומיומיים. למשל, אם יש צורך בפיטורין במפעל (לא עלינו...), הרי העובד הראשון שיפלט הוא האחרון שהתקבל למפעל. מכיוון שתוכנו של מצביע המחסנית הוא כתובת, הרי אורכו הוא 16 סיביות ברוב המיקרומחשבים.

2.8 אוגר מורה (Index register; IX)

אוגר זה משמש כאוגר עזר לקביעת כתובת תא בזכרון בשיטת גישה מיוחדת אשר נקראת מיעון בעזרת אוגר מורה (INDEXED ADDRESSING; X), עליה עוד נדון בסעיף 6.4.5, ויש לו גם שימושים נוספים. אורכו כאורך הכתובת, כלומר 16 סיביות. במיקרומעבדים מסויימים לא קיים אוגר מורה.

2.9 אוגרים לשימוש המתכנת

האוגרים לשימוש המתכנת הם:

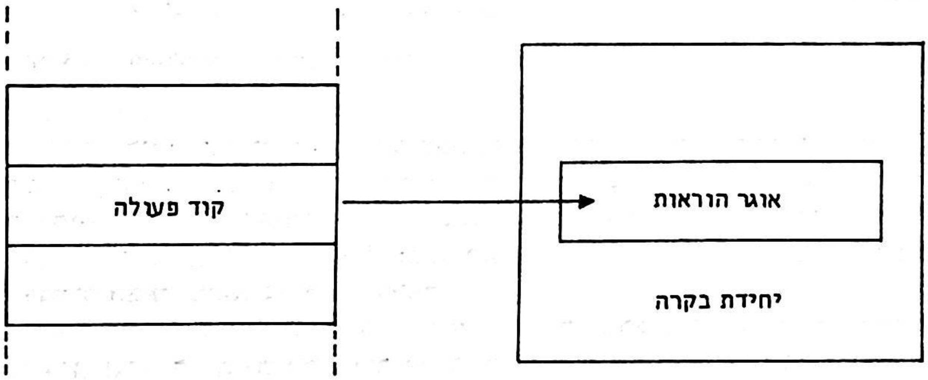
צובר A	ACCUMULATOR A
צובר B	ACCUMULATOR B
אוגר הדגלים	FLAGS REGISTER
מונה תוכנית	PROGRAM COUNTER
מצביע מחסנית	STACK POINTER
אוגר מורה	INDEX REGISTER

המתכנת נעזר רבות באוגרים אלה בתכנות.

2.10 תהליך ביצוע ההוראות במחשב

הוראה מורכבת מאחת או יותר מילים, כאשר המילה הראשונה מהווה את קוד הפעולה (ראה סעיף 1.7 לתזכורת). את תהליך ביצוע ההוראה ניתן לחלק לשניים:

א. שלב ההבאה (FETCHING) הוא שלב העתקת קוד הפעולה של ההוראה אל יחידת הבקרה (ראה איור 2.6). המטרה היא לטעון את קוד הפעולה באוגר ההוראות (IR) כדי לאפשר ליחידת הבקרה לפענח את מהות ההוראה.

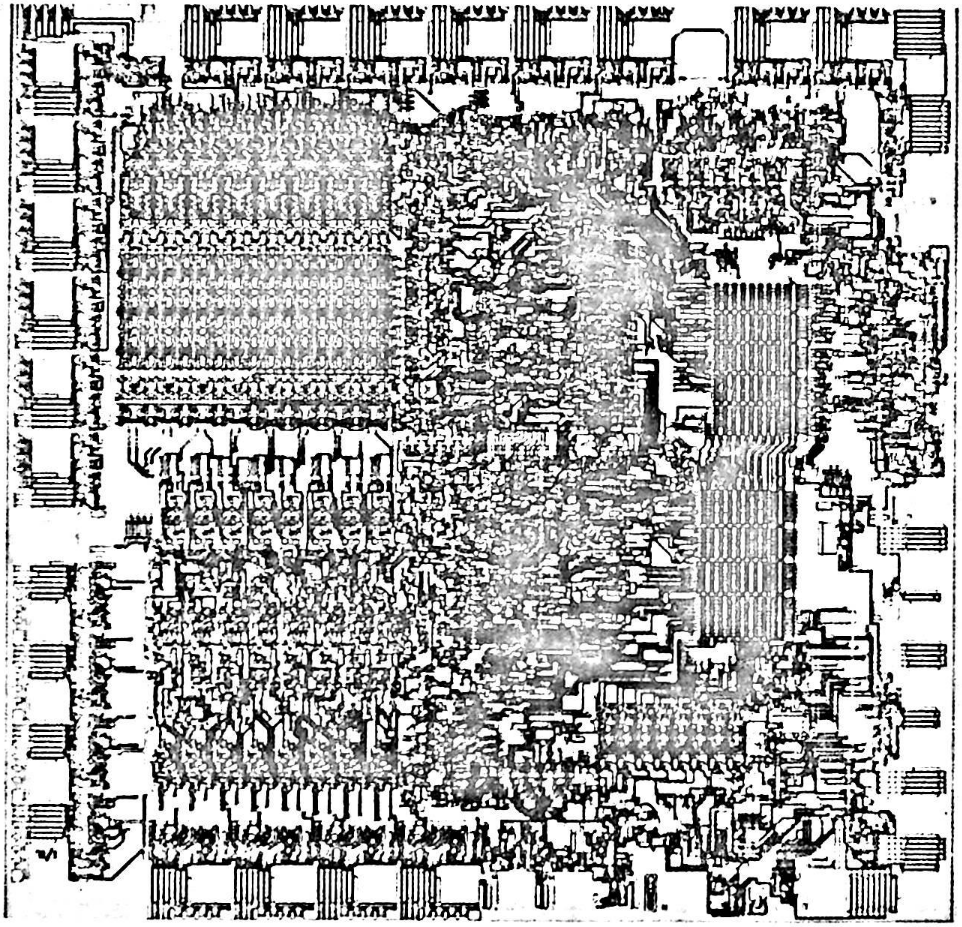


איור 2.6 - שלב ההבאה (FETCHING) של הוראה

ב. שלב הביצוע (EXECUTION) מתקיים לאחר פענוח קוד הפעולה. יחידת הבקרה יוזמת, מפעילה, מתזמנת ומבקרת את ביצוע ההוראה. שלב הביצוע עשוי להכיל סדרת פניות לזכרון או להתקני קלט/פלט על מנת לקרוא נתונים או לכתוב תוצאות. פעולות המתרחשות בשלב הביצוע הן בהתאם לקוד הפעולה של ההוראה. בשלב זה עשויים להשתתף אוגרים בשימוש המתכנת והאוגר הזמני.

2.11 משך הזמן לביצוע הוראה

לביצוע הוראה דרושה סדרת פעולות קריאה (READ) או כתיבה (WRITE) בזכרון. פעולת קריאה "מעתיקה" נתון מכתובת מסויימת אל היע"מ. פעולת כתיבה "מעתיקה" תוצאה (נתון) בכיוון ההפוך, מהיע'מ אל הכתובת כנדרש. פעולה כזאת נמשכת בפרק זמן הנקרא מחזור מכונה (MACHINE CYCLE). מחזור מכונה מורכב ממחזורי שעון המערכת (CLOCK CYCLE). קיימים מיקרומעבדים שבהם מחזור המכונה שווה למחזור השעון כפי שזה קיים ב-6800. מחזור השעון נמדד בסדר גודל של עשיריות המיקרושניה.



מיקרומעבד אינטל 8080

2.12 שאלות חזרה

1. שרטט תרשים מלבנים של מיקרומעבד כלשהו והסבר תפקידו של כל מלבן.
2. שרטט מבנה פנימי של מיקרומעבד כלשהו והסבר בקצרה כל מרכיב בו.
3. אוגר ההוראות –
א. מהו תפקידו?
ב. מדוע לדעתך הוא שייך ליחידת הבקרה?
ג. נניח כי ההוראה העומדת להתבצע יש בה שלוש מילים:
01110010 10010101 00111001
מה בדיוק יאוחסן באוגר ההוראות?
ד. האם יש למתכנת שליטה על אוגר הוראות?
4. מונה תוכנית –
א. מה תפקידו?
ב. מדוע הוא מתוכנן להתקדם באחד (בפעולתו הרגילה)?
ג. מה יקרה אם המונה יתקדם בשניים?
ד. האם אפשרית נסיגה?
ה. האם הרצף 0111000110 עשוי להיות תוכנו של מונה תוכנית אופיני?
5. מצביע מחסנית –
א. על מה הוא מצביע?
ב. מהו סדר התקדמותו לצורך הכנסת המידע למחסנית?
ג. הסבר את המונח LIFO.
6. אוגר מורה –
א. מה הוא מורה?
ב. האם הוא אופיני לכל סוג של מיקרומעבד?
7. יחידה אריתמטית לוגית –
א. מה הם תפקידיה?
ב. הסבר כל צובר ואוגר בהם נעזרת היחידה.
ג. כמה צוברים יש לכל הפחות במיקרומעבד?
ד. איזו מילה (או מילים) מההוראה שבשאלה 3ג' תגיע ליחידה?
8. שלבי הרצת ההוראה –
א. תאר שלב הבאת ההוראה
ב. תאר שלב ביצוע ההוראה
ג. האם תוכנו של אוגר הוראות עשוי להשתנות תוך שלב הביצוע? מדוע?
ד. האם תוכנו של מונה תוכנית עשוי להשתנות תוך שלב הביצוע? מדוע?
ה. האם תוכנם של צוברים ואוגר זמני עשויים להשתנות תוך שלב הביצוע? מדוע?

פרק 3 - שפות תכנות

מטרות ביצועיות לימודיות

עם סיום העיון בפרק זה הקורא יוכל:

1. להסביר את עקרון הוראת מחשב ואת שפת המכונה
2. למנות בעיות הקשורות בעבודה עם שפת המכונה והשימוש בקוד הקסדצימלי
3. למנות ולהסביר לפחות שתי שיטות תרגום מקידוד הקדצימלי לשפת המכונה
4. להסביר מהי שפה הסף ואת היתרונות והחסרונות לעומת שפת המכונה
5. להסביר שיטת העבודה עם מערכות מיקרומחשב לימודית
6. להסביר מהי שפת עילית ואת היתרונות והחסרונות שלה לעומת שפת הסף
7. למנות מגמות עתידיות לגבי חומרה ותוכנה וחשיבות לימוד שפת הסף
8. להסביר מונחים חדשים המופיעים בפרק זה.

מונחים בפרק זה:

(תוכנית) טעינה הקסדצימלית	HEXADECIMAL LOADER
כינוי הוראה	INSTRUCTION CODE MNEMONICS
שפת הסף	ASSEMBLY LANGUAGE
תוכנית תרגום שפת הסף לשפת המכונה	ASSEMBLER
תוכנית המשתמש	USER PROGRAM
תוכנית המקור	SOURCE PROGRAM
תוכנית היעד	OBJECT PROGRAM
שפת עילית	HIGH-LEVEL LANGUAGE
בייסיק	BASIC
פורטרן	FORTRAN
תוכנית תרגום שפה עילית לשפת מכונה	COMPILER
תוכנית פיענוח שפה עילית לשפת מכונה	INTERPRETER

3.1 מבוא

בפרק 1 הזכרנו את המושגים הוראה, תכנות ושפת מכונה. מושגים אלה שייכים לתחום התוכנה (SOFTWARE) של המחשב. בפרק זה נרחיב מעט את היריעה ונדון ברמות תכנות שונות ונבחן שפות תכנות מיצגות מבחינת נוחות המשתמש והיבטים כלכליים.

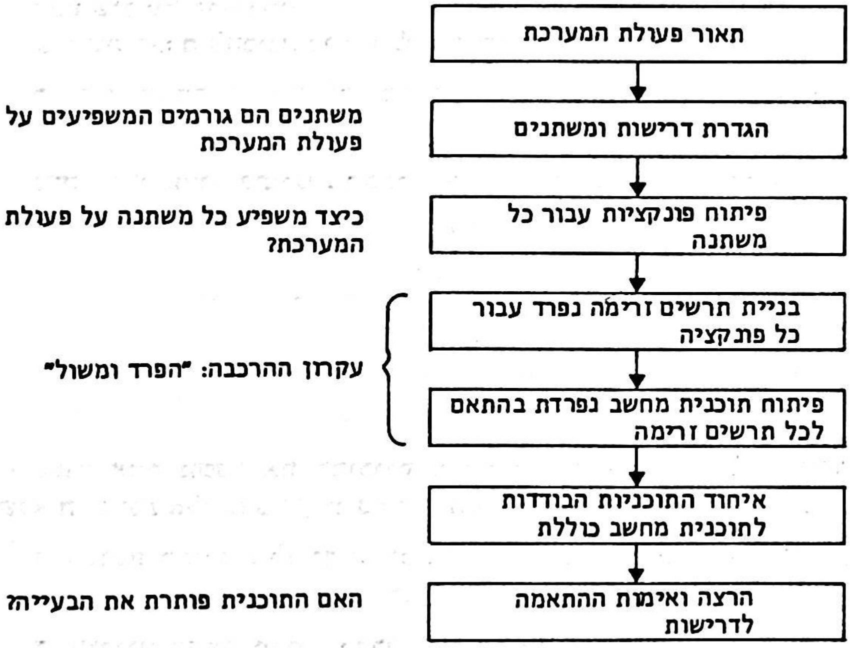
דוגמאות אחדות בפרק זה מתייחסות אמנם למיקרומעבד מסוג 6800 אולם אין הדבר גורע מכלליות הדיון, שכן העקרונות הינם משותפים לכל סוגי הציוד הממוחשב.

3.2 משמעות ההוראה ושלבי פיתוח תוכנית מחשב

מילת הוראה היא רצף מוגדר של סיביות בינריות במילת מחשב. ברצף זה חייבת להופיע סיבית לכל קו מקוי הנתונים של היע'מ על מנת שיובן כהוראה. לדוגמא, כאשר בשלב ההבאה (FETCHING) יופיע הרצף 01001111 בקוי הנתונים של היע"מ מסוג 6800 הוא יובן כהוראה האומרת "אפס צובר A".

באופן דומה הרצף 10000110 יובן כך: "טען צובר A בנתון שכתובתו תופיע במילת המחשב הבאה". מעבד מכל סוג שהוא מקבל מילה בינרית כרצף רמות לוגיות אשר מגדיר חד-חד-משמעית את קוד הפעולה או כתובת או נתון. מעבד אינו מקבל מילה המכילה מספרים בשיטת ספירה דצימלית, אוקטלית או הקסדצימלית.

רצף הוראות חד משמעיות הממצות את כל האפשרויות מוגדר כתוכנית. באיור 3.1 ניתן תרשים זרימה כללי לפיתוח תוכנית מחשב



איור 3.1 - שלבי פיתוח תוכנית מחשב

3.3 תוכנית מחשב בשפת המכונה

תוכנית מחשב היא סידרת הוראות אשר גורמת למחשב לבצע סדרת פעולות אריתמטיות או לוגיות.

הבה ניקח תוכנית לדוגמא שתלווה אותנו לכל אורך הפרק הנוכחי. התוכנית תגרום למיקרומעבד 6800 לחבר נתון כלשהו הנמצא בתא הזכרון שכתובתו 96 עם נתון נוסף הנמצא בתא הזכרון שכתובתו 97. את תוצאות החיבור יש לאחסן בתא הזכרון שכתובתו 98. להלן התוכנית:

הוראה ראשונה	10010110 01100000
הוראה שניה	10011011 01100001
הוראה שלישית	10010111 01100010

זאת היא תוכנית בשפת מכונה (MACHINE LANGUAGE). כל סיבית 1 מייצגת רמת מתח (רמה לוגית) גבוהה (HIGH); כל סיבית 0 מייצגת רמת מתח (רמה לוגית) נמוכה (LOW).

3.4 בעיות התכנות בשפת מכונה

תיכנות בשפת מכונה עלול לגרום לקשיים רבים לאדם המתכנת. מה הם? הבט שוב על התוכנית:

א. קשה לאדם להבינה כי אין לה כל משמעות סמלית עבורו.

ב. קשה לאתר בה טעויות. מספרים בינריים נראים דומים מאוד, במיוחד אם עובדים פרקי זמן ארוכים.

נסה לאתר טעות בתוכנית הבאה, אשר אמורה להיות זהה לקודמת:

10010110
01100000
10011011
01110001
10010111
01100010

טעות אחת משנה את התוכנית מקצה לקצה. המחשב עלול לבצע דברים שלא התכוונת אליהם כלל, או במקרה הטוב, לא לבצע מאומה!

ג. הכנסת התוכנית לציוד הקלט הינה איטית מאוד, כי עליך להזין (בהקשה במקלדת) כל סיבית וסיבית בנפרד.

ד. תוכנית בשפת מכונה בדרך כלל ארוכה מאוד. לכן האדם עלול לטעות פה ושם ולאבד זמן רב לצורך תיקונים, כפי שכבר נוכחת לדעת זה עתה.

3.5 כתיבה בשיטת ספירה הקסדצימלית במקום בינרית

זהו רעיון המנצל את העובדה שכל ארבע סיביות בינריות ניתן לתרגם בקלות לסיפרה הקסדצימלית אחת, ולהיפך. לצורך רענון אופן התרגום ראה נספח א'.

התוכנית שלנו תראה עתה כך:

96
60
9B
61
97
62

נסה כעת למצוא הטעות:

96
60
9B
71
97
62

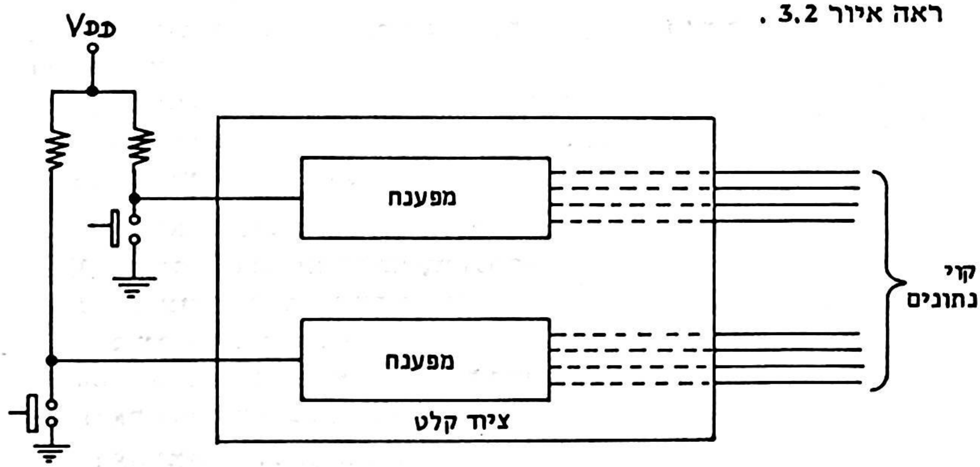
המלאכה הינה הפעם קלה ומהירה בהרבה. גם מלאכת ההזנה של תוכנית הקסדיצימלית (מעתה נסמן ה.ד.) תגזול זמן קצר בהרבה. אך איך נזין תוכנית ה.ד. בשעה שהמחשב קולט רק סיביות בינריות?

3.6 טעינה הקסדצימלית

במיקרומחשבים הנפוצים כיום מלאכת התרגום מתבצעת על ידי המחשב עצמו באחת משתי אפשרויות:

א. חומרה - כל סיפרה ה.ד. מיוצגת על ידי מקש במקלדת. מעגלי הפענוח של פעולת ההקשה מתרגמים את פעולת הקליד לסידרה של סיביות אשר מייצגות את הסיפרה הבינרית.

ראה איור 3.2 .



איור 3.2 - תרגום קוד הקסדצימלי לבינרי על ידי חומרה

ב. תוכנה - התרגום מתבצע באמצעות תוכנית לסריקת המקשים (HEXA-DECIMAL LOADER) הנמצאת בזכרון באופן קבוע. כל מקש מייצג ספרה ה.ד. והתוכנית מתרגמת לחיצה על מקש לארבע סיביות בינריות מתאימות שתופענה בארבעה קוי נתונים. השיטה המקובלת בכל המיקרומחשבים היא שיטת התרגום על ידי תוכנה אשר משיגה פשטות בחומרה על חשבון תפיסת חלק מהזכרון לצורך התוכנית.

3.7 שפת הסף

השימוש בבסיס ה.ד. מפשט את תהליך הכנת התוכנית והכנסתה למיקרומחשב, אבל אין בה עדיין פתרון לבעייה ההבנה. כלומר, זמן מה אחרי שכתבנו את התוכנית אנו שוב מתקשים להבין מה היתה בעצם מהות התוכנית. נסה לקרוא שוב את התוכנית שבסעיף 3.5 . אתה רואה רשימת מספרים שאינם מייצגים עבור הקורא כל משמעות לוגית.

שפת הסף או שפת אסמבלי (ASSEMBLY LANGUAGE) "מרככת" בעייה זו במידה מסויימת, כי היא מאפשרת:

א) לתת כיטוי לקוד הפעולה של הוראה (INSTRUCTION CODE MNENOMICS). למשל, LDAA STAA, ADDA.

ב) לקבוע כיטוי כראשי תיבות של ביטוי אנגלי המסביר את משמעות ההוראה. למשל:

טען צובר A	LDAA – LoaD Accumulator A
חבר צובר A	ADDA – ADD accumulator A
אחסן צובר A	STAA – STore Accumulator A

ג) להרחיב את ההוראה במילה נוספת ועל ידי שימוש באות נוספת להגדיר חד משמעית:

(1) אם המילה הנוספת היא נתון או כתובת;
(2) מהיכן יש להעתיק את הנתון או את הכתובת;
(3) כמה מילים בסך הכל יש בהוראה.

למשל, האות הנוספת D בקוד הפעולה, משמעותה:

(1) המילה הבאה היא כתובת (מקוצרת);
(2) שממנה יש להעתיק את הנתון;
(3) בהוראה יש שתי מילים.

התוכנית שלנו תיראה אם כן בשפת הסף כך:

הוראה ראשונה - LDAA D, 60
הוראה שניה - ADDA D, 61
הוראה שלישית - STAA D, 62

אם נרשום עתה את התוכנית בשפת הסף לצד התוכנית בשפת המכונה בשיטת ספירה ה.ד. נקבל את התמונה הבאה:

LDAA D, 60	96
	60
ADDA D, 61	9B
	61
STAA D, 62	97
	62

התוכנית שנכתבה קודם עבור המיקרומעבד 6800 תיכתב כך עבור מיקרומעבד מסוג 8085:

```
LDA   60 H
MOV   BA
LDA   61 H
ADD   B
STA   62 H
```

3.8 סיכום ביניים

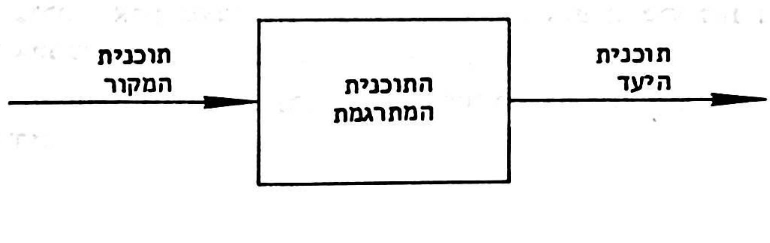
עבור כל המערכות הבסיסיות של מיקרומחשב הנמצאות כיום בשימוש מערכת החינוך הטכנולוגי בארץ כותבים תוכניות בשפת הסף ומכינים אותן להזנה למיקרומחשב בשפת המכונה בשיטת הספירה ההקסדצימלית.

תוכנית באסמבלי היא עבורנו תוכנית המשתמש (USER PROGRAM) או תוכנית המקור (SOURCE PROGRAM). תוכנית בשפת המכונה היא תוכנית היעד (OBJECT PROGRAM).

3.9 תוכנית תרגום לשפת מכונה

זה עתה דנו בשפת הסף ובהמשך נזכיר גם שפות נוספות, אולם המחשב מבין שפת המכונה הבינרית בלבד. כדי שנוכל לפנות למחשב בשפה שאינה שפת המכונה יש צורך בתוכנית תרגום המאוחסנת באופן קבוע בזכרון. ראה איור 3.3.

תוכנית המתרגמת משפת הסף נקראת אסמבלר (ASSEMBLER).



איור 3.3 – מנגנון התרגום על ידי תוכנה הקבועה בזכרון

3.10 שפות עיליות

לשפת הסף יש יתרון בולט לעומת שפת המכונה, כי אפשר להבין אותה. לעומת זאת יש לה חסרונות שהבולטים בהם:

א. תוכניות בשפת הסף הינן ארוכות בדרך כלל.

ב. שפת הסף איננה ניידת. כלומר, לכל מיקרומעבד שפת הסף שלו ואי אפשר להעביר תוכנית בשפת סף ממחשב אחד לשני.

שפת עילית (HIGH LEVEL LANGUAGE) מכוונת לענות על חסרון זה. למשל, תוכנית החיבור שלנו בשפת בייסיק (BASIC) תראה כך:

LET C = A + B

תוכנית זו תראה כך בשפת פורטרן (FORTRAN):

SUM = NUMB1 + NUMB2

בשני המקרים השתמשנו בסמלי המשתנים ולא בהוראות או בסימולים השייכים למחשב מסויים בלבד. לתוכנית בשפת עילית יתרונות חשובים נוספים:

א. תוכנית בשפה עילית קצרה בהרבה לעומת זו בשפת האסמבלי.

ב. שפה עילית הינה ניידת בין מחשבים, אולם צריך מיד להוסיף: במידה רבה אך לא באופן מלא. גם עבור שפה עילית קיימת זיקה כלשהי לסוג המעבד, בעיקר בהוראות המפעילות תכונות מסויימות של החומרה. למשל, עבור ציוד מתוצרת APPLE או COMMODORE (מיקרומעבד מסוג 6502 או 6510). קיימות הוראות בשפת בייסיק כגון PEEK או POKE כאשר המקבילות להן בציוד מתוצרת RADIO SHAKE (מיקרומעבד Z -80) הן ההוראות SET או RESET. קיימים הבדלים חשובים נוספים אשר לא נעסוק בהם עתה.

ברור מכל מקום שבעזרת שפה עילית ניתן לכתוב תוכניות בפשטות ובמהירות הגבוהה בסדר גודל לעומת כתיבה בשפת אסמבלי. אולם, גם לשפות עיליות יש חסרונות, כאשר החשובים בהם:

א. התוכנית המתרגמת (COMPILER) שפה עילית לשפת מכונה תופסת נפח זכרון גדול באופן יחסי לזכרון הכולל של מערכת מיקרומחשב כיום. לדוגמא, 20K תאים מתוך 64K הרי הם כשליש מכלל נפח הזכרון. תוכנית מתרגמת מסוג שונה היא התוכנית המפענחת (INTERPRETER), המתאימה לתרגום משפת בייסיק. היא חסכונית יותר, אבל גם לה חסרונות. מכאן נובע, כי נפח הזכרון מגביל שימוש בשפות עיליות במיקרומחשב.

ב. שפות עיליות אינן מאפשרות ניצול כל תכונות היע"מ ביעילות כפי שיכולה שפת אסמבלי.

ג. דרושים מרכיבים נוספים למערכת הן בחומרה והן בתוכנה, דבר המייקר מאוד את מחירה.

3.11 מה לגבי העתיד?

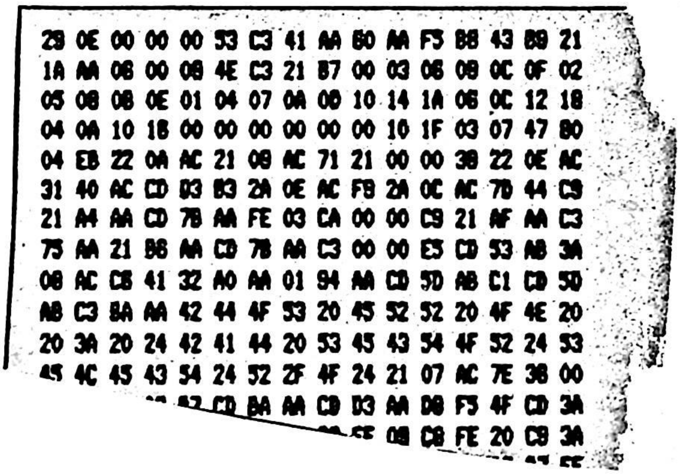
יש להניח כי העתיד נתון לשפות העיליות. הסיבות הבולטות לכך הן:
(א) ניתן להתקין זכרון גדול בנפח הולך וקטן;
(ב) מחיר החומרה הולך ונעשה זול יותר;
(ג) עלות המתכנתים הולכת ונעשית גבוהה יותר;
(ד) שפות עיליות תהיינה יותר ויותר תקניות ולכן ניידות.

3.12 מדוע להשתמש בשפת הסף?

אם בעתיד נשתמש בשפות עיליות, מדוע נלמד היום שפת אסמבלי ושפת מכונה?

להלן הסיבות העיקריות:

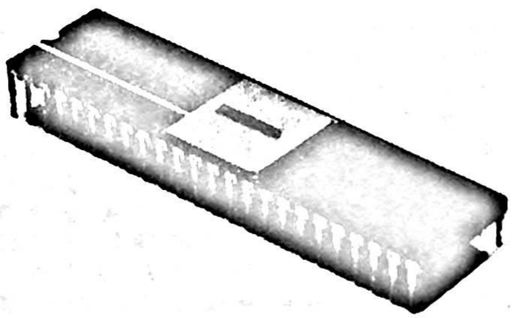
א. עבור שימושים רבים דרושה היעילות של שפת אסמבלי היורדת ל"קרביים" של המעבד ומנצלת את מלוא הגמישות והאפשרויות הטמונות בו.
ב. בתעשיה יש שימוש רב מאוד לשפת הסף במיקרומחשבים המותקנים במכשירים שונים.
ג. ידע מעמיק של שפת הסף מסייע לפיתוח תוכנה בשפות עיליות ולייעול התיכנות בשפות אלו.



תוכנית מחשב בקוד הקסדצימלי

3.13 שאלות חזרה

1. מהם הקשיים הכרוכים בתכנות בשפת מכונה?
2. מהי מהות השיפור של קידוד הקסדצימלי?
3. מהם יתרונות שפת הסף לעומת שפת מכונה?
4. כיצד נכתוב ונריץ תוכנית בעזרת קיט המיקרומחשב המצוי?
5. מהם היתרונות של שפה עילית לעומת שפת הסף?
6. מהם החסרונות של שפה עילית לעומת שפת הסף?
7. מדוע נחוץ לדעת שפת הסף?



פרק 4 - מערכת מחשב בסיסית עם מיקרומעבד 6800

4.1 מבוא
4.2 מבנה מערכת בסיסית עם מיקרומעבד 6800
4.3 זכרון לקריאה ולכתיבה RAM
4.4 זכרון לקריאה בלבד ROM
4.5 מישק תקשורת מקבילית PIA
4.6 מישק תקשורת טורית אסינכרונית ACIA
4.7 מישק תקשורת טורית סינכרונית SSDA
4.8 שאלות חזרה

מטרות ביצועיות לימודיות

עם סיום העיון בפרק זה הקורא יוכל:

1. לשרטט דיאגרמת מלבנים של מערכת מחשב בסיסית עם מיקרומעבד מסוג 6800, וכו'
2. להסביר בקצרה תפקידו של כל מרכיב במערכת המחשב הבסיסית.
3. להבחין בין מערכת בסיסית למערכת מינימלית.
4. לתת לפחות שלוש דוגמאות של ציוד קלט מקבילי.
5. לתת לפחות שלוש דוגמאות של ציוד קלט פלט טורי.
6. להסביר מונחים חדשים בפרק זה.

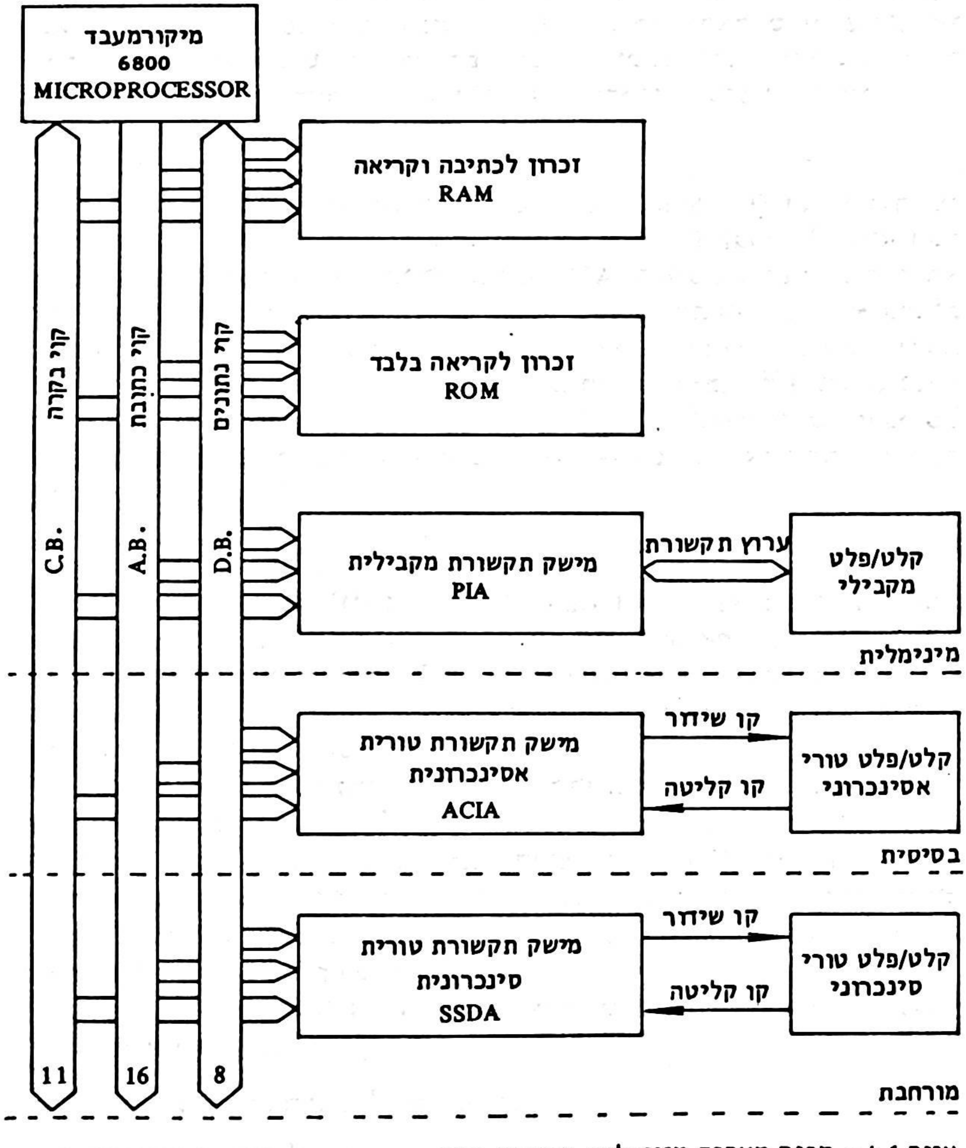
מונחים בפרק זה:

זכרון לקריאה ולכתיבה	RANDOM ACCESS MEMORY; RAM
זכרון לקריאה בלבד	READ ONLY MEMORY; ROM
מישק תקשורת מקבילית	PERIPHERAL INTERFACE ADAPTOR; PIA
מישק תקשורת טורית אסינכרונית	ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTOR; ACIA
מישק תקשורת טורית סינכרונית	SYNCHRONOUS SERIAL DATA ADAPTOR; SSDA
זכרון למשתמש	USER MEMORY
משגוח תוכנית הפעלה	MONITOR
ניפוי (שגיאות)	DEBUGING
תוכנית העורך	EDITOR

4.1 מבוא

עד כה עסקנו בתאור מערכת מחשב אופיינית, לאו דוקא מיוחדת. החל מפרק זה ואילך נדון במערכת מיקרומחשב עם היע'מ מסוג 6800. נעזר בחומר שהוזכר עד כה, אולם נעסוק בעיקר במרכיבי סוג זה של מערכת, ובמשך הזמן תווכח בודאי לדעת, כי קיים קשר ודמיון רב בין הכללי לפרטי. בדרך זו נציג מערכת בסיסית ונתאר בקצרה את תפקידי מרכיביה השונים. בפרק 7 נעסוק במרכיבים אלה באופן מפורט יותר.

4.2 מבנה מערכת בסיסית עם מיקרומעבד 6800



איור 4.1 – מבנה מערכת מינימלית, בסיסית ומורחבת

מיקרומעבד 6800 או יע"מ הינו המרכיב העיקרי במערכת ואליו מתחברים במקביל שאר מרכיבי המערכת באמצעות שלושת סוגי הקוים: נתונים, כתובות ובקרה. מילת המחשב היא בת שמונה סיביות ולכן קוי הנתונים הם שמונה במספר. ניתן אם כן ליצור עד $256 = 2^8$ צירופים אפשריים, כלומר, עד 256 מילים שונות. קוי הנתונים הם דו-כיווניים על מנת לאפשר זרימת מידע אל ומחוץ ליע"מ.

קוי הכתובות הם 16 במספר, כלומר, כתובת מורכבת משתי מילים. ניתן אם כן ליצור עד $65536 = 2^{16}$ כתובות שונות. נהוג לבטא את המספר ב-K ‏($2^{10} = 1024$) ועל כן לפנינו אפשרות לפנות ל-64K כתובות, קוי הכתובות הם חד כיווניים בלבד אשר יוצאים מהיע'מ אל מרכיבי המערכת. קוי הבקרה הם 11 במספר כולל כניסות השעון. כזכור (ראה סעיף 2.2), קוי הבקרה אינם בחינת גוש אחד, בחלקם קוים נכנסים ובחלקם יוצאים. לא כולם מגיעים לכל מרכיבי המערכת, אלא כל קו פונה לרכיב המערכת לפי יעודו.

4.3 זכרון לקריאה וכתיבה

הזכרון לקריאה וכתיבה (Random Access Memory; RAM) הוא מערך של תאים רבים שכל אחד מהם כולל 8 דלגלגים (FLIP FLOP; F.F) לכן הוא זוכר מילת מחשב אחת. לכל תא ניתן לגשת (ACCESS) באמצעות כתובתו. בכל תא ניתן לאחסן מילה אחת או להעתיקה ממנו. פעולת האחסון בתא היא פעולת כתיבה (WRITE), אשר מוחקת אוטומטית את תוכנו הקודם. פעולת העתקה מתא היא פעולת קריאה (READ), אשר משאירה את תוכנו **ללא שינוי**. זכרון לקריאה ולכתיבה מיועד למשתמש (USER MEMORY) לצורך אחסון זמני של תוכניות, נתונים, כתובות, תוצאות. תוכן התאים אינו נשמר לאחר הפסקת הזנת המתח (הפסקת חשמל).

4.4 זכרון לקריאה בלבד

זכרון לקריאה בלבד (Read Only Memory; ROM) הוא מערך של תאים רבים כאשר בכל אחד מהם ניתן לאחסן מילת מחשב אחת, ולכל אחד ניתן לגשת באמצעות כתובתו. ההבדל החשוב בין שני סוגי הזכרון מתבטא בכך שזכרון לקריאה בלבד מיועד לקריאה בלבד, כלומר, אין אפשרות לכתוב בו ישירות על ידי תוכנית המשתמש.

זהו זכרון קבוע, שאינו נמחק עם הפסקת ההזנה, מכיוון שתכנו נקבע על ידי תהליך אלקטרוני מיוחד, תהליך הצריבה. בזכרון לקריאה בלבד היצרן צורב את תוכנית המשגוח או תוכנית הפעלת המערכת (MONITOR) הסורקת ציוד קלט/פלט. ניתן לצרוב בזכרון זה תוכניות בעלות אופי קבוע בהתאם ליעוד המערכת, לדוגמא תוכנית המטפלת בבקרת מכשיר מסויים. תוכניות צרובות למטרות לימודיות עשויות לכלול:

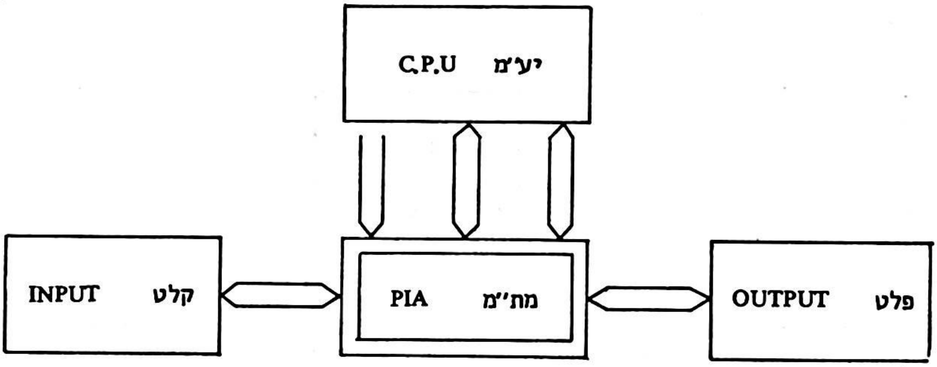
א. תוכנית העורך (EDITOR) לשם בדיקת תוכנית המשתמש לצורך ניפוי שגיאות (DEBUGING).

ב. תוכניות שרות בהתאם ליעוד המערכת.

ג. תוכניות ספריה להדגמות או למשחקים ועוד.

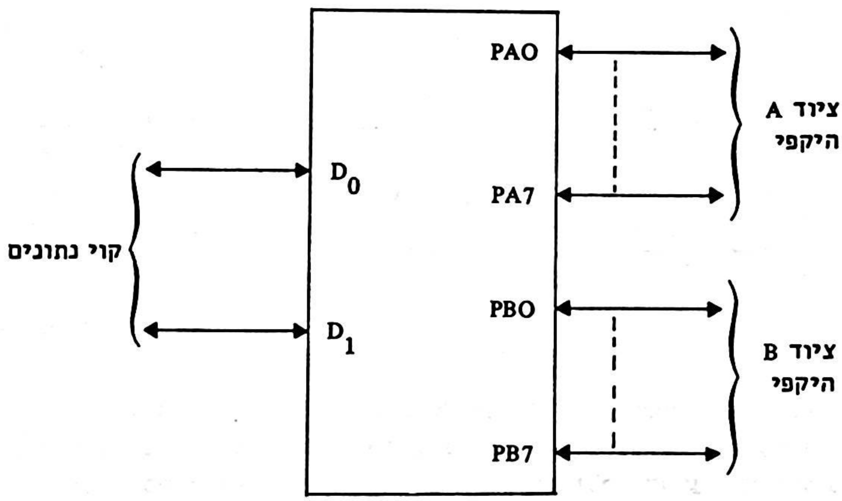
4.5 מישק תקשורת מקבילית, מת"מ

מישק תקשורת מקבילית (Peripheral Interface Adaptor; PIA) הוא מתאם (ADAPTQR) שיוצר השקה (INTERFACE) בין היע"מ לבין הציוד ההיקפי (PERIPHERAL) שלה. ראה איור 4.2:



איור 4.2 – תפקידו של מישק תקשורת מקבילית

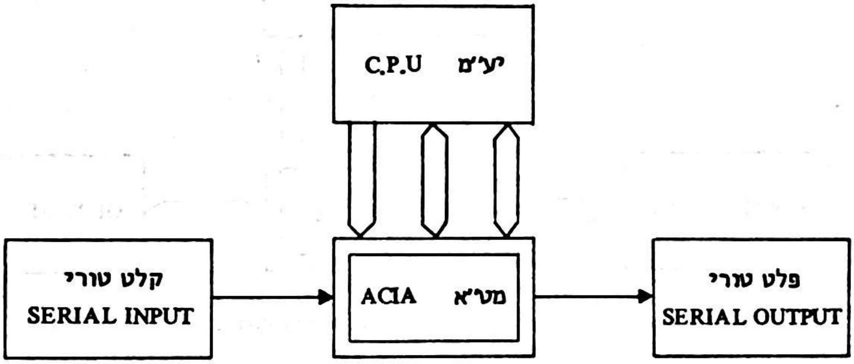
במערכת מינימלית חיוני שיהיה מת"מ אחד לפחות לצורך חיבור אל לוח המקשים שהם קלט המערכת ואל התצוגה שהיא פלט המערכת. המישק יכול לבצע גם תקשורת פשוטה, כמו לדוגמא עם רשמקול כדי שניתן יהיה להקליט תוכניות לאחסנה וגם לטען אותן שוב להפעלה. עקרון פעולת המת"מ מתואר באיור 4.3:



איור 4.3 – פעולת מישק תקשורת מקבילית

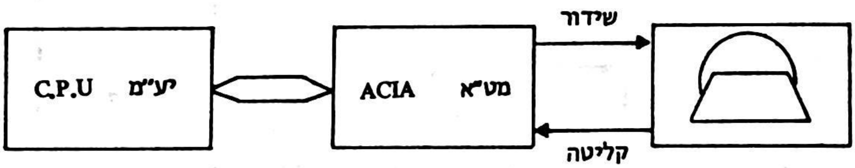
4.6 מישק תקשורת טורית אסינכרונית, מט"א

למישק תקשורת טורית אסינכרונית (Asynchronous Communications Interface Adaptor ;ACIA) תפקיד זהה לזה של מת"מ. הצורך במט"א נובע מקיומו של ציוד ק/פ טורי אשר קולט או משדר מידע בטור סיבית אחר סיבית, ולא משדר או קולט מילת מידע בבת אחת ובמקביל כפי שפועל היע"מ, ראה איור 4.4:



איור 4.4 – תפקיד של מישק תקשורת טורית אסינכרונית

דוגמא לציוד קלט טורי הוא הטלפרינטר ודוגמא לציוד פלט טורי היא מדפסת טורית (קיימת גם מדפסת מקבילית). דוגמא נוספת לשימוש במט"א היא תקשורת טלפונית עם המחשב. ראה איור 4.5:



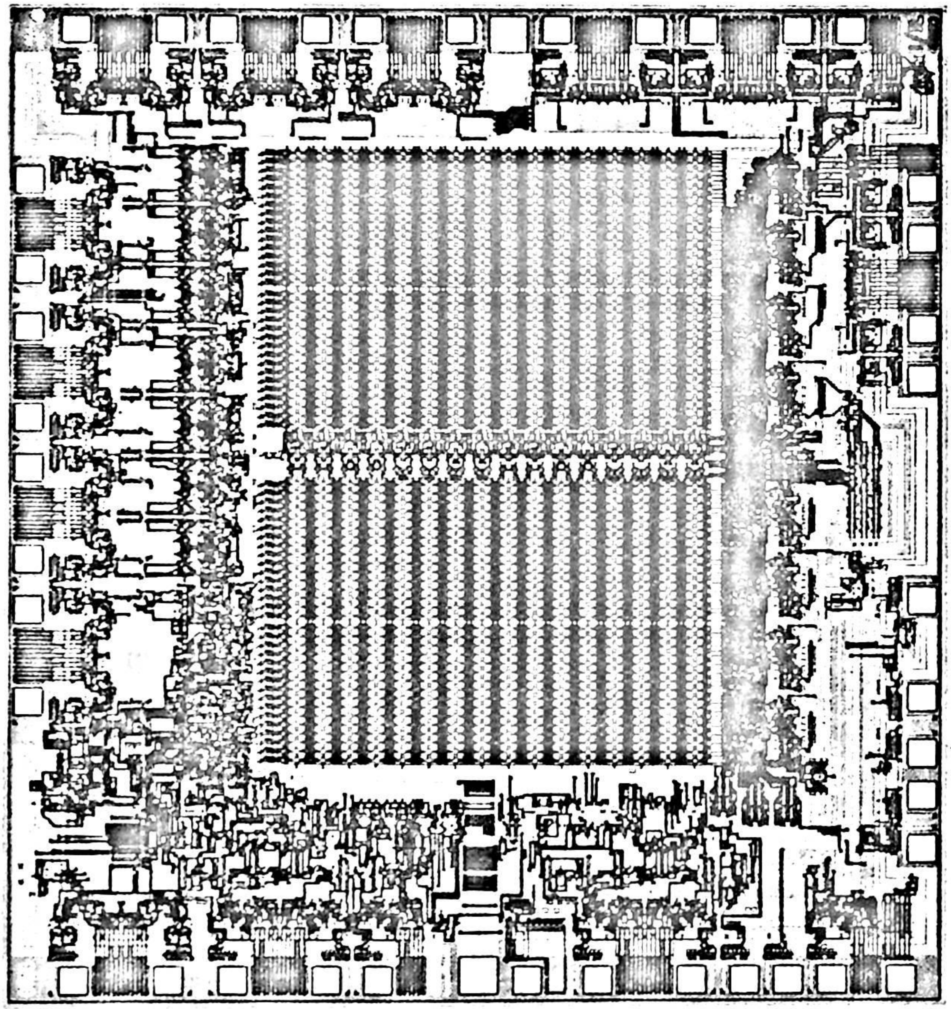
איור 4.5 – תקשורת טלפונית עם מחשב

התקשורת בין מט"א ליע"מ מתבצעת באמצעות קוי הנתונים, כבשאר מרכיבי המערכת. מכאן שעל המישק לתרגם מידע המגיע מציוד קלט למידע מקבילי עבור המערכת ולהיפך, עליו לתרגם מידע מקבילי המגיע מהמערכת למידע טורי עבור ציוד הפלט. המישק ניתן לתיכנות להשגת גמישות בתקשורת.

4.7 מישק תקשורת טורית סינכרונית, מט"ס

מישק תקשורת טורית סינכרונית (Synchronous Serial Data Adaptor SSDA) מאפשר תקשורת טורית סינכרונית, כלומר קצב מתוזמן של העברת המידע. המישק ניתן לתכנון ועל ידי כך משיגים גמישות רבה בהפעלת הציוד המבוקר כגון התאמת הקצב לסוג הציוד המבוקר, התאמת אורך המילה, בקרת שידור, בקרת קליטה ועוד. שידור וקליטה יכולים להתבצע בו זמנית (DUPLEX) בשני כיוונים בקצב תקשורת שנמדד במאות אלפי סיביות בשניה.

תקשורת טורית סינכרונית מיושמת בעבודה עם כונן דיסקטים או דיסקים ובמערכות בקרה ספרתית. המט"ס אינו בשימוש בציוד תרגול אופייני במעבדת לימוד.



זכרון אינטל 8255 RAM

4.8 שאלות חזרה

1. שרטט תרשים מלבנים של מערכת מחשב בסיסית עם מיקרומעבד מסוג 6800 והסבר בקצרה כל מלבן.

2. במה מתבטא השכלול של מערכת בסיסית לעומת מינימלית ושל מערכת מורחבת לעומת בסיסית?

3. באילו סוגים של זכרון יש להשתמש כדי לאחסן תוכנית ההפעלה, תוכנית העורך, תוכנית המתכנת?

4. מערכת פיקוח ממוחשבת של בנק מסויים מאפשרת גישה אליה לפקידי סניפי הבנק, וללקוחות מכל רחבי הארץ. ניתן לברר יתרת חשבון, להכניס או להוציא מזומנים, להפקיד המחאות, לתת הוראות קניה ומכירה לבורסה וכן שרותים נוספים. אנו מניחים, כי התקשורת בין מחשב מרכזי לבין פקידי הסניפים מבוצעת באמצעות שמונה קוי נתונים. התקשורת עם הלקוחות עשויה להיות גם באמצעות קו טלפון.

   אילו סוגי ציוד קלט ופלט דרושים:

   א. כדי להקים קשר עם פקידי הסניפים?

   ב. כדי להקים קשר עם לקוחות הבנק?

5. מישק תקשורת מקבילית, PIA –

   א. לאיזה סוג תקשורת מתאים המישק?

   ב. כמה קוים דרושים לצורך קיום תקשורת מקבילית?

   ג. כמה מכשירים יכול לשרת מישק אחד בתקשורת מקבילית?

   ד. האם ניתן להפעיל מישק זה גם לתקשורת טורית פשוטה?

6. מישק תקשורת טורית אסינכרונית, ACIA –

   א. לאיזה סוג תקשורת מתאים המישק?

   ב. כמה קוים דרושים לצורך קיום תקשורת טורית?

   ג. הבא שלוש דוגמאות לשימוש בתקשורת טורית אסינכרונית.

7. מישק תקשורת טורית סינכרונית, SSDA –

   א. מה פרוש הסינכרון?

   ב. מהי תקשורת בו זמנית?

   ג. הבא שלוש דוגמאות לשימוש בתקשורת טורית סינכרונית.

פרק 5 – דגלים ושיטות מיעון במיקרומעבד 6800

מטרות ביצועיות לימודיות

עם סיום עיון בפרק זה הקורא יוכל:

1. לפרט, לתאר ולהסביר את אופן פעולתם של כל אחד משׁשׁת האוגרים לשימוש המתכנת ביע"מ 6800,
2. לפרט, לתאר ולהסביר את אופן פעולתו של כל אחד משׁשׁת הדגלים,
3. לפרט ולתאר את כל אחת משׁש שיטות המיעון,
4. לערוך השוואה בין שיטות המיעון של ההוראה מבחינת אורך ההוראות וזמן הביצוע.

מונחים בפרק זה:

המונחים בפרק זה מופיעים ברשימת הסעיפים.

5.1 מבוא

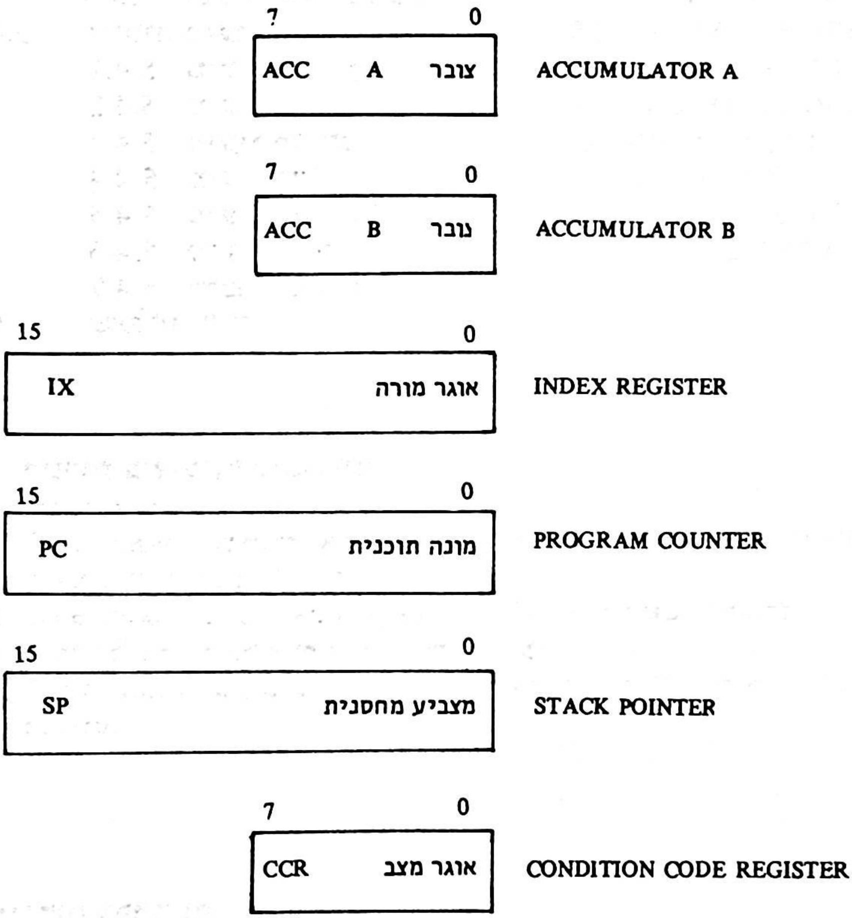
בפרק 3 הובא דיון כללי על אפשרויות התכנות, הצגנו את השפות השונות וערכנו השוואה ביניהן. בפרק הנוכחי ובפרק הבא נעסוק בשפת הסף ושפת המכונה של מיקרומחשב הבנוי מרכיבי משפחת ה-6800.

נכיר את הצוברים והאוגרים הפנימיים בתוך היע"מ בהם נעזר במלאכת התכנות. נדון באפשרויות הפניה השונות לנתונים וכתובות ונערוך השוואה ביניהם.

למיקרומעבד 6800 קיימות 72 הוראות שונות ושש שיטות איתור נתונים וכתובות, והוא משתמש ב-197 סוגי פעולה.

5.2 אוגרים בשימוש המתכנת

ביע"מ קיימים אוגרים שונים לשימוש המערכת. לששה מהם יכול המתכנת לפנות כדי לבדוק ולשנות את תוכנם באמצעות הוראות. אלה הם האוגרים למשתמש (USER REGISTERS). ראה איור 5.1:



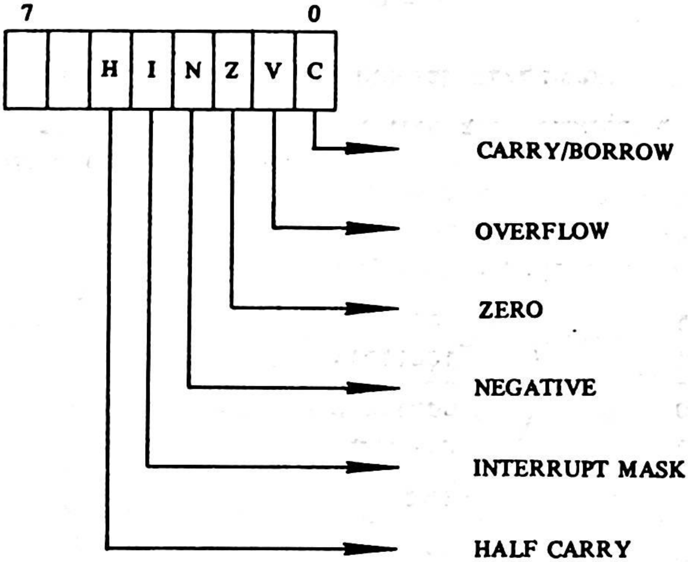
איור 5.1 – אוגרים למשתמש ביע"מ 6800

צובר A, צובר B ואוגר מצב הם בני 8 סיביות, מכאן שהם מכילים מילה אחת כל אחד. שאר האוגרים הם בני 16 סיביות, והם מכילים שתי מילים כל אחד כי כל כתובת היא בת 2 מילים. תאור האוגרים השונים למעט אוגר מצב, ניתן בסעיפים 2.4, 2.6, 2.7 ו-2.8, והם זהים עבור כל מיקרומעבד בו הם מופיעים. אוגר המצב מתואר בסעיף הבא.

נוח, קל ומקובל לתאר מספרים בינריים כמספרים הקסדצימליים (ה.ד.). לכל 4 ספרות בינריות סיפרה הקסדצימלית אחת. עם מעט תירגול הופך המעבר בין שני בסיסי הספירה למידי בשני הכיוונים. לכן מעתה ואילך נציין בדרך כלל מספרים בבסיס הקסדצימלי (ה.ד.). במקרה של ספק מספר עשרוני נציין זאת במפורש על ידי הסימן 10 ליד המספר.

5.3 אוגר מצב (Condition Code Register).

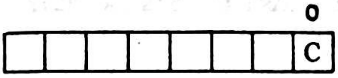
אוגר המצב הוא אוגר הדגלים (FLAGS) אשר סיביותיו מפורטות באיור 5.2:



דגל הנשא	CARRY/BORROW
דגל ההצפה	OVERFLOW
דגל האפס	ZERO
דגל הסימן השלילי	NEGATIVE
דיל הפסיקה	INTERRUPT MASK
דגל נשא למחצה	HALF CARRY

איור 5.2 - דגלים באוגר מצב

לשתי הסיביות המשמעותיות ביותר אין תפקיד מוגדר. שתיהן נמצאות בדרך כלל במצב 1 לוגי, אבל המשתמש יכול לשנותן. מנגנון אוגר המצב בנוי כך שדגל מורם (כלומר, במצב 1 לוגי) מסמל שכתוצאה מביצוע ההוראה האחרונה התרחשה התופעה שהדגל מייצגה. נבהיר את הדבר בהמשך.



5.3.1 דגל הנשא (Carry/borrow; C)

דגל הנשא מורם כאשר קורה אחד מן האירועים הבאים:

א. כתוצאה מפעולת חיבור נוצר נשא, למשל:

11011010
\+ 01010111
1 00110001
נשא 1
C = 1

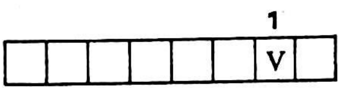
10011010
\+ 01010111
0 11110001
נשא 0
C – 0

ניתן לומר שהנשא "עובר להתאחסן" בדגל הנשא.

ב. כתוצאה מפעולת חיסור נוצר צורך בשאילה, כלומר המספר המחסר הגדול מהמחוסר. למשל:

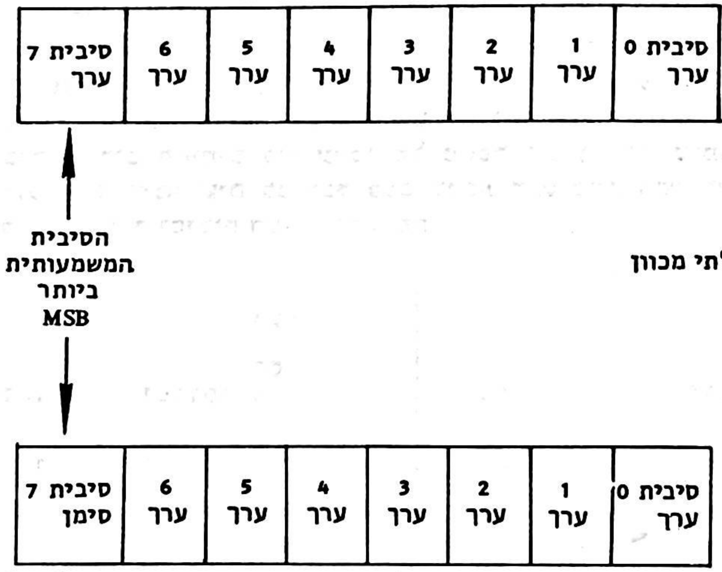
1 10110101
− 11011011
11011010
יש שאילה
C=1

0 10110101
− 10011011
00011010
אין שאילה
C=0



5.3.2 דגל ההצפה (oVerflow; v)

דגל הצפה מונף מצביע על תוצאה שגוייה בפעולה עם מספרים מכוונים. כאן נעשה אתנחתא קלה ונבהיר מספר דברים, ובתחילה נברר מהו מספר מכוון. בשם זה אנו מכנים מספר שהסיבית המשמעותית ביותר שלו (Most Significant Bit; MSB) מצביעה על סימנו (כלומר, חיובי או שלילי) ולא על ערכו. אם הסיבית היא 0 הרי המספר חיובי. אם הסיבית היא 1 הרי המספר שלילי. למשל, המספר 4+ הוא 00000100 (MSB=0), ולעומת זאת המספר 4- הוא 11111100 (MSB=1). נסכם ונאמר כך: במספר מכוון סיבית מספר 7 של מילת המחשב היא סיבית הסימן. ראה איורים 5.3 ו־5.4 .



איור 5.3 – מספר בלתי מכוון

איור 5.4 – מספר מכוון

טבלת מספרים מכוונים תמצא בנספח ג'.

בשמונה סיביות של מספר בלתי מכוון ניתן לבטא 256_{10} ערכים שונים, מ-00 עד FF. כאשר מקדישים סיבית אחת לציון סימנו של המספר נשארות רק שבע סיביות לציון הערך. אם כך, כמה מספרים מכוונים חיוביים ניתן ליצור? – 128, החל מ-0 עד $127_{(10)}$ או מ-00 עד 7F הקסדצימלי, או מ-00000000 עד 01111111 בינרי. אגב, שים לב לכך שעל פי ההגדרה, אפס הוא מספר חיובי. כמה מספרים מכוונים שליליים ניתן ליצור? – 128, החל מ-1- ועד $128_{(10)}$ – או מ-FF עד 80 הקסדצימלי, או מ-11111111 עד 10000000 בינרי.

עתה הגיע הרגע להיזכר ברעיון המשלים לשניים שלמדת זה מכבר. משלים לשניים של מספר נתון הוא ערכו השלילי של מספר זה. ומהו המשלים לשניים של המספר השלילי? הוא אינו אלא ערכו החיובי של מספר זה ! למשל:

המספר 4 מהדוגמה הקודמת:	00000100
משלים ל-1:	11111011
	+ 1
המספר 4- הוא המשלים ל-2 של 4+	11111100
נבצע אותה פעולה על	
המספר 4-. משלים ל-1:	00000011
	+ 1
חזרה למספר 4+ !	00000100

ומהי התועלת משימוש במספרים מכוונים? אם המחשב יודע לבצע פעולת חיבור, הרי שבעזרת מספרים מכוונים הוא יכול לבצע גם חיסור (A) – (B) על ידי חיבור (A) + (–B). בצורה זו אפשר להציג חיסור על ידי חיבור! אלא שיש כאן בעייה, כי בפעולות על מספרים מכוונים עלולות להיווצר לעיתים שגיאות בגלל ערוב תחומים בין ערכו של מספר מכוון לבין סימנו. נעיין במספר דוגמאות, ואנא שים לב בכל פעם לנשא מהסיביות השישית והשביעית שהן סיבית הערך וסיבית הסימן בהתאמה:

		נשא	
		00	
109	+	01101101	+
–117		10001011	
–8		11111000	

		נשא	
		11	
–53	+	11001011	+
109		01101101	
56		00111000	(1)

משתי הסיביות אין נשא — תוצאות נכונות — משתי הסיביות יש נשא

מסקנה: התוצאה תמיד נכונה אם בשתי הסיביות המשמעותיות מתקבל נשא זהה, נשא 00 או נשא 11.

ודוגמא נוספת:

		נשא	
		10	
–19	+	11101101	+
–119		10001001	
? 118		01110110	(1)

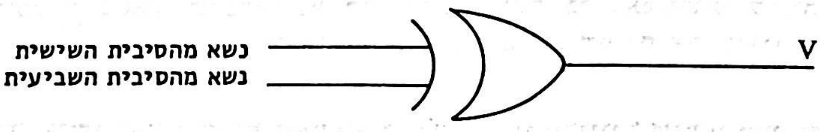
		נשא	
		01	
75	+	01001011	+
109		01101101	
? –72		10111000	

נשא שונה משתי הסיביות (10) תוצאות שגויות נשא שונה משתי הסיביות (01)

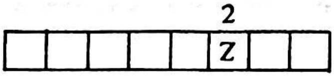
מסקנה: התוצאה תמיד שגויה אם מתקבל נשא שונה משתי הסיביות המשמעותיות ביותר. הוכחת טיעון זה היא מעבר למטרות ספר זה. תוכל לבדוק זאת במספר דוגמאות משלך.

אולם, מה לכל העניין הזה ולדגל ההצפה (V)? אוגר המצב בודק נשא מהסיביות השישית והשביעית בתוצאת הפעולה על מספרים מכוונים. אם הנשא שונה, הרי התוצאה שגויה והדגל V המורם "1" מציין זאת. אחרת – התוצאה נכונה והדגל מוסר "0".

אגב, האם אתה מעלה בדעתך בזה הרגע כיצד אפשר לבצע בדיקה כזאת? לתשובה ראה איור 5.5:

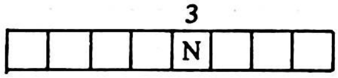


איור 5.5 - מעגל עקרתי של דגל ההצפה



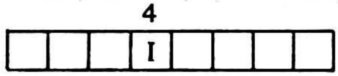
5.3.3 דגל האפס (Zero; Z)

דגל האפס מורם כאשר תוצאת הפעולה היא אפס (00000000).



5.3.4 דגל הסימן השלילי (Negative; N)

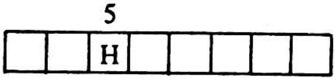
דגל הסימן השלילי מורם כאשר תוצאת הפעולה היא *שלילית*. אוגר המצב בודק בתוצאה את הסיבית השביעית שהיא הסיבית המשמעותית ביותר עבור מספר מכוון, ואם היא 1 המספר הוא שלילי. מסיבה זו אפשר לקרוא לדגל הזה בשם דגל הכיוון.



5.3.5 דגל הפסיקה (Interrupt; I)

הפסיקה משרתת בעיקר את יחידות קלט/פלט של המחשב.

באמצעות דגל הפסיקה מודיעה היע"מ למרכיבי המערכת שברגע זה מתבצעת תוכנית כלשהי ואין להפריע לה. אם יחידת קלט/פלט מבקשת שרות שמשמעותו הרצת תוכנית אחרת, יש להמתין עד שיוסר הדגל. למנגנון הפסיקות של 6800 יש דרגות עדיפות שונות. תוכנית הפסיקה העדיפה ביותר אינה מתחשבת במצב הדגל ותאלץ את היע"מ לבצעה בכל מקרה. פרוט מלא של מנגנון הפסיקה ראה בסעיף 6.10 ובפרק 7.



5.3.6 דגל הנשא למחצה (Half Carry; H)

דגל הנשא למחצה פעיל כאשר מתבצעות פעולות על מספרים המיוצגים בצופן BCD. הדגל מורם כאשר יש נשא מהסיבית השלישית. צופן זה, כזכור לך בודאי, מבטא כל סיפרה עשרונית מ-0 עד 9 באמצעות ארבע סיביות בינריות. מכאן ברור כי הסיבית השלישית היא המשמעותית ביותר, ולכן תפקודי דגל הנשא למחצה זהים לאלה של דגל הנשא. במילת מחשב אחת בת 8 סיביות ניתן לאחסן שתי ספרות עשרוניות המיוצגות בצופן BCD.

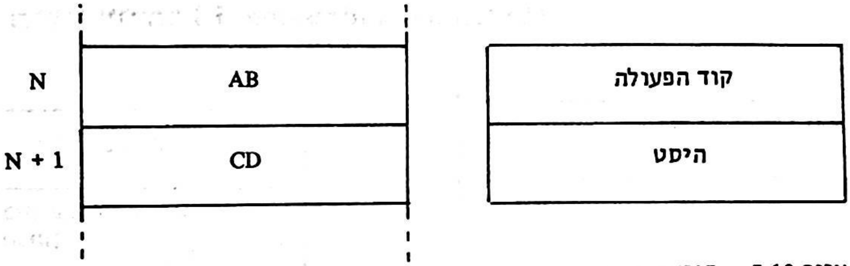
5.4 שיטות מיעון (ADDRESSING MODES)

מהו מיעון? כל תהליך עיבוד נתונים מלווה בתנועת הנתונים אל היע"מ וממנה. הנתונים עשויים להימצא באחד מתאי הזכרון, באוגרים פנימיים של היע"מ, במישק לתקשורת מקבילית (PIA) ובמישק לתקשורת טורית (ACIA או SSDA). כדי לשלוף או לאחסן מידע בתא מסויים יש לנקוב בכתובתו שהיא מענו המדויק. אם כן מיעון הוא הגדרת כתובת.

הנתון נמצא בתא שכתובתו מצויינת באופן חלקי בהוראה, בשמונת סיביותיה הפחות משמעותיות. מכאן שהוראה במיעון ישיר הינה בת שתי מילים ותופסת שני תאי זכרון סמוכים. האם על המחשב לנחש מהי המילה הראשונה של הכתובת? לא ! היא חייבת להיות 00. מכאן נובע שעל הנתון להמצא באחד מ-2^8 התאים הראשונים בזכרון בלבד בין 0000 לבין 00FF (255 דצימלי), לדוגמא 007E. ראה איור 5.9 .

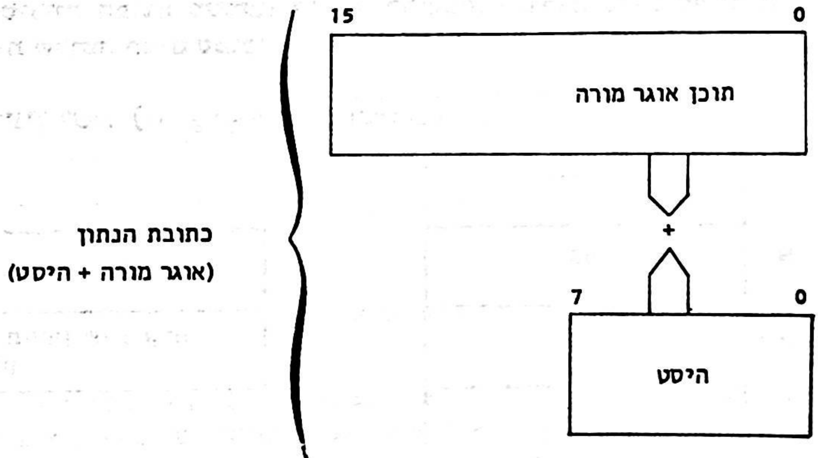
מיעון ישיר מהווה שיכלול של המיעון המורחב. הוא חוסך תא בזכרון, משום שההוראה כוללת רק שתי מילים, וחוסך גם זמן ביצוע ההוראה לעומת המיעון המורחב. אולם, אין יתרונות בלבד. חסרון השיטה מתבטא במגבלת מרחב הכתובות של הנתון אשר יכול להימצא באחד מ-255 התאים הראשונים בזכרון בלבד.

5.4.5 מיעון באמצעות אוגר מורה (indeXed addressing; X)



איור 5.10 - הוראה באמצעות אוגר מורה

להוראה מעין זו שתי מילים ולכן היא תופסת שני תאים עוקבים בזכרון, כפי שמוצג באיור 5.10. שש-עשרה הסיביות של אוגר מורה מהוות כתובת בסיסית. ההוראה מספקת היסט (OFFSET) כלומר, מספר בין 00 לבין FF שיש להוסיף לתוכן אוגר מורה על מנת לקבל את כתובת הנתון. ההיסט הוא תמיד לפנים.

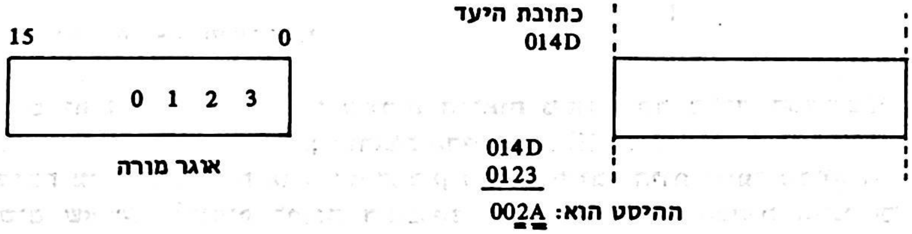


איור 5.11 - יצירת כתובת הנתון במיעון באמצעות אוגר מורה

מיעון באמצעות אוגר מורה מהווה שילוב של מיעון מורחב עם מיעון ישיר, והוא מאפשר להגיע לכל תא בזכרון.

כיצד קובע המתכנת את ההיסט עבור מיעון באמצעות אוגר מורה? להלן הנוסחה "הגואלת" (ראה איור 5.12):

- "כתובת היעד" פחות "כתובת הנמצאת באוגר מורה" שווה להיסט.
- "כתובת היעד" היא הכתובת בזכרון שאליה יש להגיע.
- "כתובת הנמצאת באוגר מורה" היא הכתובת שנטענה לאוגר מורה מבעוד מועד.



איור 5.12 – הדגמת חישוב היסט במיעון באמצעות אוגר מורה

יש לקחת בחשבון את האילוצים הבאים:

א. את ההיסט מוסיפים תמיד ואין אפשרות להפחית. אם יש צורך לפנות אל תא שכתובתו נמוכה מזו הרשומה באוגר מורה, דרוש לשנות את תוכן האוגר לפני כן.

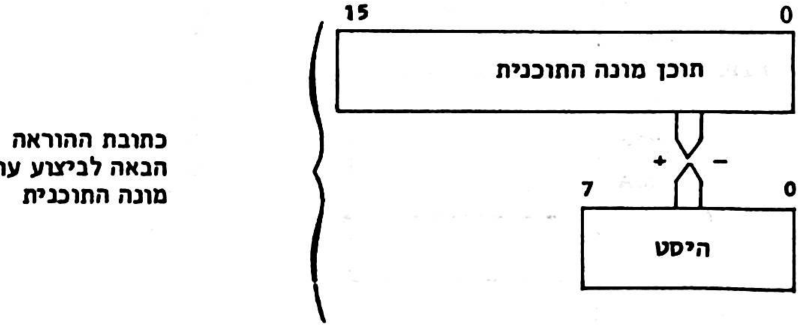
ב. תמיד יש לציין את ההיסט, גם אם הוא 00.

ג. תוכן האוגר עצמו אינו משתנה על ידי הוספת ההיסט וניתן להשתמש בו שוב ושוב ככתובת יחוס.

ד. ההיסט המירבי הוא FF (255 דצימלי).

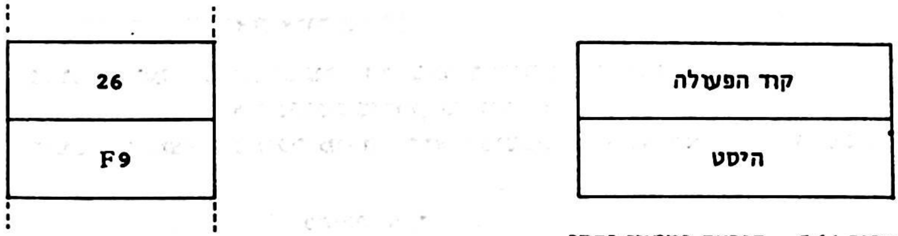
5.4.6 מיעון יחסי (Relative addressing; R)

מיעון יחסי מגדיר כתובת כשאר שיטות המיעון. אך שלא ככל השאר, המיעון היחסי מגדיר כתובת של הוראה ולא של נתון. לעיתים ברצוננו להפנות את המשך ביצוע התוכנית לפקודה שאינה ברצף ההוראות (ראה סעיף 2.6 לצורך רענון הנושא). כדי לעשות זאת, יש לשנות את הכתובת שבמונה התוכנית. ראה איור 5.13:



איור 5.13 – יצירת כתובת ההוראה הבאה על ידי מיעון יחסי

הכתובת שנוצרה זה עתה תאוחסן במונה התוכנית והוא ימשיך להתקדם החל מכתובת זו צעד אחרי צעד. להוראה במיעון יחסי שתי מילים: קוד הפעולה והיסט ולכן היא תופסת שני תאים סמוכים בזכרון. ראה איור 5.14.



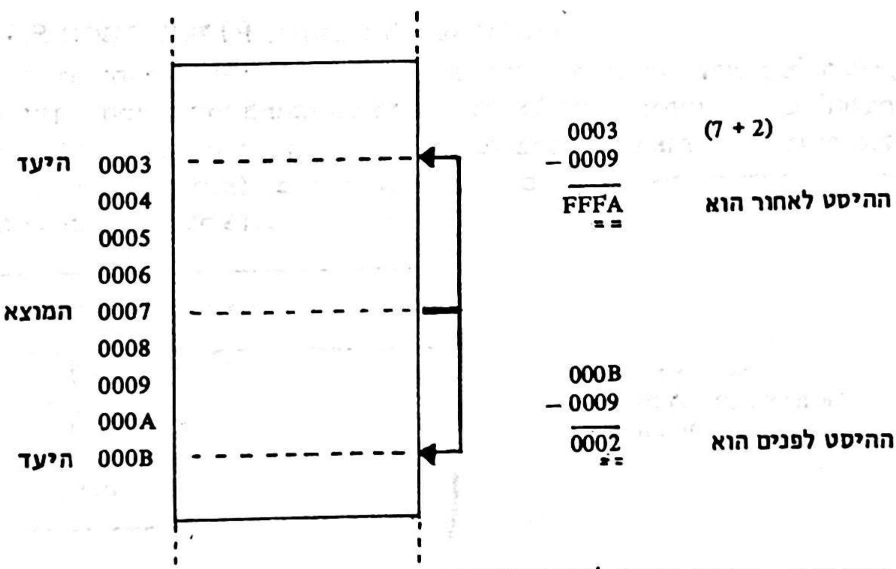
איור 5.14 - הוראה במיעון יחסי

בסעיף 6.7 יתברר לך כי קיימות הוראות מסוג אחד בלבד המשתמשות במיעון יחסי, ורק בו. אלו הן הוראות ההסתעפות (BRANCH). בניגוד להיסט בשיטת מיעון באמצעות אוגר מורה, כאן ההיסט חייב להיות מספר מכוון. זאת משום שאפשר להוסיף למונה התוכנית עד 127_{10} או להפחית ממנו עד 128_{10}. כלומר, ניתן לדלג על הכתובות בשני הכיוונים.

כיצד קובע המתכנת את ההיסט במיעון יחסי? להלן הנוסחה:

- "כתובת היעד" פחות "כתובת המוצא ועוד שניים" שווה להיסט,
- "כתובת המוצא" היא הכתובת בזכרון שבה נמצא קוד הפעולה של הוראת ההסתעפות,
- "כתובת היעד" היא הכתובת בזכרון שבה נמצא קוד הפעולה של הוראה שאליה יש להגיע.

באיור 5.15 מוצגת דוגמא של היסט לפנים ולאחור. כתובת המוצא היא 7 וכתובות היעד הן B ו-3 בהתאמה. ראה גם טבלת עזר לחישובי היסט בנספח ג'.

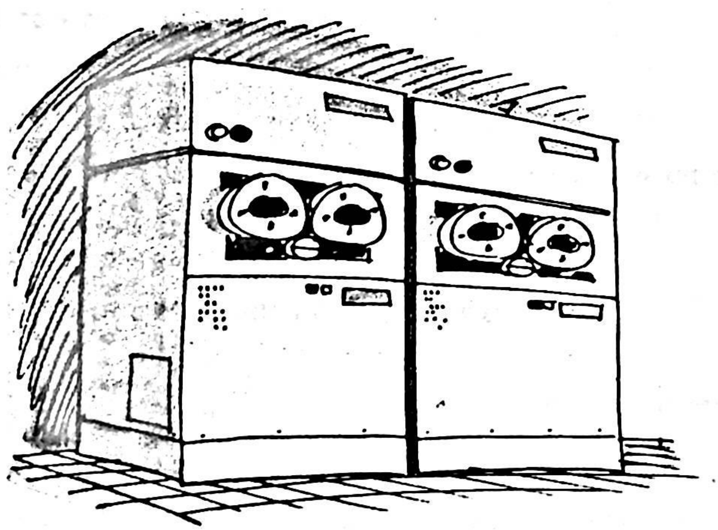


איור 5.15 - שימוש בהיסט לשם הסתעפות

5.4.7 סיכום שיטות המיעון

נסכם בטבלה תכונות אחדות של שיטות המיעון השונות:

סוג המיעון	מספר מילים בהוראה	הוראה כוללת קוד הפעולה ועוד...
מובלע	1	
מיידי I	2	נתון
מורחב E	3	כתובת מלאה של נתון
ישיר D	2	כתובת מקוצרת של נתון
אוגר מורה X	2	היסט לצורך כתובת הנתון
יחסי R	2	היסט מכוון לצורך כתובת ההוראה הבאה



"זכור את דברי - לא ירחק היום
והמיקרומחשבים יתפשו את מקומנו"

5.5 שאלות חזרה

הערה: יש להעזר בנספח א' לפי הצורך.

1. מה תפקידו של אוגר מצב?
2. האם יונף דגל הנשא כתוצאה מהפעולות הבאות:
   א. 11100010 + 01110001
   ב. 11001100 - 01100110
3. האם יונף דגל ההצפה כתוצאה מהפעולות הבאות:
   א. 01111000 + 00111010
   ב. 11000011 + 11000110
4. האם יונף דגל האפס כתוצאה מאיפוס הצובר A?
5. מה תפקידו של דגל הפסיקה?
6. האם יונף דגל הנשא למחצה כתוצאה מהפעולות הבאות:
   א. 11000101 + 00111011
   ב. 01010011 + 11000110
7. אילו דגלים יונפו כתוצאה מהפעולות הבאות:
   א. 11000101 + 10101101
   ב. 00110101 + 00101111
8. באיזה סוג מיעון משתמשת ההוראה הקצרה ביותר?
9. באיזה סוג של מיעון משתמשת ההוראה הארוכה ביותר?
10. ערוך השוואה בין אפשרויות ההיסט המותר עבור מיעון יחסי לעומת מיעון באמצעות אוגר מורה.
11. מהו יתרון המיעון הישיר לעומת המיעון המורחב?
12. האם ניתן להגיע לתא בזכרון בעזרת המיעון המובלע? מדוע?

פרק 6 - תכנות למיקרומעבד 6800

מטרות ביצועיות לימודיות

עם סיום העיון בפרק זה הקורא יוכל:

1. למנות ולהסביר סוגי ההוראות של 6800
2. לנתח תוכנית נתונה בשפת הסף או בשפת המכונה בקוד הקסדצימלי
3. לכתוב ולהריץ תוכניות על פי דרישות מוגדרות בשפת הסף ובשפת המכונה בקוד הקסדצימלי,
4. להבחין בין הוראת דילוג להוראת הסתעפות,
5. להבחין בין הוראות התת-שיגרה להוראות הפסיקה,
6. להסביר שלבי ביצוע ההוראה, לפסיקת תוכנה,
7. לחשב זמן ביצוע תוכנית נתונה,
8. להבחין בין תוכנית חסכונית בזמן הביצוע לבין תוכנית חסכונית מבחינת נפח הזכרון.

מונחים בפרק זה:

LOAD; LD	טען
STORE; ST	אחסן
TRANSFER; T	העבר
CLEAR; CLR	אפס
INCREMENT; INC	הגדל ב-1
ADD	חבר
AND	בצע פעולת "גם" לוגי
COMPLEMENT; COM	השלם ל-1
EXCLUSIVE OR; EO	בצע פעולת "XOR" לוגי
SET INTERRUPT; SEI	הרם דגל הפסיקה
LOGIC SHIFT RIGHT; LSR	הזזה לוגית ימינה
ARITHMTIC SHIFT RIGHT; ASR	הזזה אריתמטית ימינה
ROTATE RIGHT; ROR	סובב ימינה
JUMP; JMP	דלג
BRANCH ALWAYS; BRA	הסתעף תמיד
BRANCH IF EQUAL ZERO; BEQ	הסתעף אם דגל האפס מורם
BRANCH IF MINUS; BMI	הסתעף אם דגל הכיוון מורם
PUSH; PSH	הפעל
PULL; PUL	משוך
BRANCH TO SUBROUTINE; BSR	הסתעף לתוכנית משנה
JUMP TO SUBROUTINE; JSR	דלג לתוכנית משנה
RETURN FROM SUBROUTINE; RTS	חזור מתוכנית משנה
SOFTWARE INTERRUPT; SWI	פסיקת תוכנה
WAIT FOR INTERRUPT; WAI	המתן לפסיקה
RETURN FROM INTERRUPT; RTI	חזור מפסיקה
NO OPERATION; NOP	אל תבצע דבר

6.1 מבוא

ליחידת עיבוד מרכזית 6800 קיימות 72 הוראות יסודיות. כל הוראה ניתן להפעיל בשיטות מיעון אחדות מבין שש השיטות האפשריות. אם נסכם את כל המיעונים המותרים לכל הוראה והוראה, נגיע לסך של 197 קודי פעולה.

בפרק הנוכחי מסווגות ההוראות ב-10 קבוצות בהתאם לאופיין. לקבוצה נתונה קוים משותפים רבים כפי שנראה בדוגמאות שתוצגנה בכל קבוצת הוראות. אין כל צורך לזכור על פה את כל ההוראות, אך צריך לדעת להבחין ביניהן ולהשתמש בטבלאות השונות שרוכזו בנספח א' לפי מפתחות איתור שונים.

פרק זה הינו מבוא לתכנות ולעבודה מעשית עם המיקרומחשב לצורך אימות העקרונות ולהרחבת הידע בתכנות 6800.

נערוך השוואה בין גישות שונות לפתרון בעיות בתכנות. נדון בכלים הדרושים על מנת להחליט מהי התוכנית היעילה לפתרון בעיה נתונה מבחינת החסכון בנפח הזכרון וחסכון בזמן הדרוש לביצוע.

6.2 הוראות טעינה, אחסון והעברת נתונים

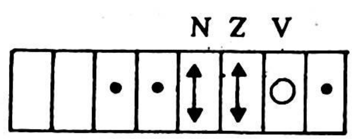
טען - (LoaD; LD). הוראה לטעינת אוגר בנתון שיועתק מזכרון.
אחסן - (STore; ST). הוראה לאחסון תוכן האוגר בזכרון.
העבר - (Transfer; T). הוראה להעברת נתונים מאוגר לאוגר.

הוראות אלה משפיעות על התאים בזכרון, הצוברים והאוגרים כמפורט:
צובר A, צובר B, תא זכרון M, אוגר מורה IX, מציין מחסנית SP, אוגר מצב CCR.

את ההשפעה על הדגלים נסמן כך:
↕ הדגל מושפע ומתייצב בהתאם לתוצאות ביצוע ההוראה
1 מונף (בנספח א' מסומן ב- S)
0 מוּסר (בנספח א' מסומן ב- R)
● הדגל לא מושפע

עבור כל הוראה נציין שיטות המיעון האפשרית בה.
(1) טען צובר LoaD Accumulator LDA

אוגר מצב



פעולה	מורחב E	אוגר X	ישיר D	מיידי I	מיעון / כינוי
M→A	B6	A6	96	86	LDAA
M→B	F6	E6	D6	C6	LDAB

נניח, כי עלינו לטעון צובר A בנתון 89. נבדק כיצד נוכל להעזר בטבלה כדי לכתוב הוראה למחשב:

- בשפת הסף
- בשפת המכונה (הקסדצימלית)

כיצד נכתוב הוראה זו בשפת הסף? לטעינת צובר A הכיטוי הוא LDAA. הנתון 89 נמצא בהוראה עצמה ולכן נשתמש במיעון מיידי I, ולכן ההוראה בשפת הסף תהיה כתובה כך: LDAA I ,89. סימן הפסיק מפריד בין קוד ההפעלה לבין הנתון (89) והוא מרמז על שתי מילים בהוראה.

וכיצד נכתוב הוראה זו בשפת המכונה (הקסדצימלית)? המיעון הוא מיידי ולכן ברור שבהוראה יש שתי מילים. המילה הראשונה: את קוד הפעולה הנמצא בהצטלבות של שורה LDAA וטור I נרשום בהקסדצימלית ומכאן שהמילה הראשונה היא 86. המילה השנייה: הנתון עצמו, היא 89.

היכן נאחסן את ההוראה? - במרחב הזכרון הפנוי לשימושנו, בשני תאים סמוכים. נניח כי את המילה הראשונה נאחסן בתא 0710, ולכן את המילה השניה נאחסן בתא 0711. את ההוראה הכתובה בשתי השפות מקובל לרשום כך:

הוראה בשפת הסף ASSEMBLY LANGUAGE	הוראה בשפת המכונה הקסדצימלית MACHINE LANGUAGE	כתובת ההוראה INSTRUCTION ADDRESS
LDAA I ,89	86	0710
	89	0711

לאחר שהוסברה כתיבת ההוראות בשלבים, נוכל בהמשך לרשום את ההוראות בקיצור כדלקמן:

ההוראה	מידע	כתובת
LDAA I, 89	86	0710
	89	0711

ניקח דוגמא נוספת.

נניח כי יש לטעון צובר B בנתון הנמצא בתא שכתובתו היא 00A7. אם נבחר במיעון המורחב ההוראה תיראה כך:

ההוראה	מידע	כתובת
LDAB E, OOA7	F6	O2CD
	00	O2CE
	A7	O2CF

אם לעומת זאת נבחר במיעון ישיר, נוכל לחסוך תא בזכרון ולרשום את ההוראה כך:

O2CD	D6	LDAB D, A7
O2CE	A7	

בדוגמאות הבאות נוותר על ציון הכתובות של ההוראות.

(2) אחסן צובר — STore Accumulator — STA

	D	X	E	
STAA	97	A7	B7	A→M
STAB	D7	E7	F7	B→M

				N	Z	V	
		•	•	↕	↕	0	•

דוגמא: עלינו לאחסן צובר A בתא שכתובתו 1234:

0000	B7	STAA X, 1234
0001	12	
0002	34	

(3) העבר מ- A ל- B — Transfer from A to B — TAB

TAB	16

A → B

				N	Z	V	
		•	•	↕	↕	0	•

להזכירך, משבצת ריקה בטבלה מסמלת מיעון מובלע. מכיוון שבהוראה מעין זו הפעולה הנדרשת היא בין אוגרים בתוך היע"מ עצמו כלומר "בתוך הבית", אין כל צורך לנקוב בכתובת חיצונית כלשהי. מכאן המסקנה ההגיונית שלמעשה אין כל צורך במיעון נוסף על המיעון המובלע.

בשלב זה מומלץ לעבור לביצוע ניסוי בהעברת נתונים. העזר גם בנספחים א' ו-ג'.

6.3 הוראות אריתמטיות

הוראות אריתמטיות גורמות לביצוע פעולות חישוב כגון חיבור, חיסור וכו'. נעיין במספר דוגמאות של הוראות אלה.

(1) אפס   CLR   CLeaR

	X	E		
CLR	6F	7F		00→M
CLRA			4F	00→A
CLRB			5F	00→B

				N	Z	V	C
		•	•	0	1	0	0

מטרת ההוראה היא לאפס, לנקות תא בזכרון, בצובר A או בצובר B.

נעיין הפעם בסיביות אוגר המצב. מבט בוחן מוליך למסקנה שמצב הדגלים אינו שרירותי אלא הגיוני בהחלט. תוצאת הפעולה של הוראת האיפוס תגרום לכך שדגל האפס (Z) מורם. בנוסף לכך, אפס (00000000) הוא מספר שבו הסיבית המשמעותית ביותר היא אפס, ולכן מובן גם שדגל הסימן השלילי (N) יורד (כלומר - במצב 0).כאשר התא מאופס לא תתכן הצפה ולכן דגל ההצפה (V) יורד. מאותה הסיבה עצמה ברור כי דגל הנשא (C) חייב לרדת.

כדאי לעיין בסיביות אוגר המצב גם לאחר ביצוע בהוראות אחרות. נעים להווכח שכדאי להפעיל הגיון בריא, ואין כל צורך לזכור על פה כל דבר.

(2) הגדל באחד את אוגר מורה   INcrement iX   INX

INX	08

$IX + 1 \rightarrow IX$

		H	I	N	Z	V	C
		•	•	•	↕	•	•

(3) חבר תוכן תא בזכרון לצובר   ADD memory to accumulator   ADD

	I	D	X	E	
ADDA	8B	9B	AB	BB	A + M→A
ADDB	CB	DB	EB	FB	B + M→B

		H	I	N	Z	V	C
		↕	•	↕	↕	↕	↕

לדוגמא, מה משמעות ההוראה EB, 59 בשפת המכונה?

המילה הראשונה היא קוד הפעולה. קוד EB קובע שיש לחבר לצובר B נתון שנמצא בכתובת המוגדרת על ידי תוכן האוגר המורה ועוד היסט של 59 הקסדצימלי.

עתה אפשר לבצע ניסויים בכתיבת הוראות אריתמטיות. העזר גם בנספח א'.

6.4 הוראות לוגיות

הוראות לוגיות מבצעות פעולות XOR, AND, OR, משלים ל-1, משלים ל-2.
נעיין במספר הוראות מייצגות.

(1) בצע פעולת "גם" לוגי בין צובר לזכרון — AND

	I	D	X	E	
ANDA	84	94	A4	B4	A.M→A
ANDB	C4	D4	E4	F4	B.M→B

		H	I	N	Z	V	C
		•	•	↕	↕	0	•

מהו זמן הביצוע של הוראה ANDA I, 20? עיין בנספח א' והנח כי מחזור השעון הוא מיקרושניה אחת.

(2) בצע פעולת המשלים ל-1 — COMplement — COM

	X	E	
COM	63	73		$\bar{M}$→M
COMA			43	$\bar{A}$→A
COMB			53	$\bar{B}$→B

		H	I	N	Z	V	C
		•	•	↕	↕	0	1

נניח כי בתא 0100 בזכרון נמצא הנתון 01011011. כתוצאה מביצוע ההוראה COM E, 0100 יימצא בתא זה הנתון 10100100.

(3) בצע XOR לוגי בין צובר לזכרון — Exclusive OR — EOR

	I	D	X	E	
EORA	88	98	A8	B8	A ⊕ M→A
EORB	C8	D8	E8	F8	B ⊕ M→B

		H	I	N	Z	V	C
		•	•	↕	↕	0	•

נניח כי בתא 00A1 נמצא הנתון BC ובצובר A נמצא הנתון 9E. מה יהיו הנתונים בתא הזה ובצובר A לאחר ביצוע ההוראה (בשפת המכונה) הבאה:

0010 98
0011 A1

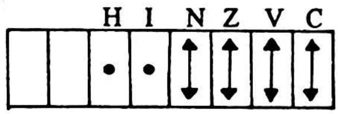
התוצאה בצובר A תהיה: 22
עתה תוכל לבצע ניסויים בתיכנות הוראות לוגיות.

6.5 הוראות אוגר מצב

הוראות אוגר מצב משפיעות באופן ישיר או בעקיפין על הדגלים באוגר זה.

(1) השווה צובר לתא בזכרון — CoMPare — CMP

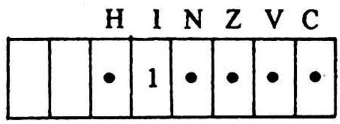
	I	D	X	E	
CMPA	81	91	A1	B1	A +M
CMPB	C1	D1	E1	F1	B +M



תוכן הצובר והתא אינם משתנים עקב ביצוע ההשוואה, כי מטרת ההוראה להשפיע על דגל אוגר המצב בלבד. נניח כי בצובר A נמצא הנתון 12 ואילו בכתובת ABCD נמצא הנתון 01. כתוצאה מביצוע ההוראה: CMPA E, ABCD יתאפסו הדגלים N, Z, V, C כי הנתון בצובר גדול מהנתון בתא. יחד עם זאת, תוכן הצובר לא השתנה והוא נשאר 12. גם תוכן התא ABCD בזכרון לא השתנה והוא נשאר 01. זהו ההבדל הבולט בין ההוראה CMP להוראת החיסור SUB (ראה נספח א').

(2) הרם דגל הפסיקה — SEt Interrupt — SEI

SEC	OF	I =1



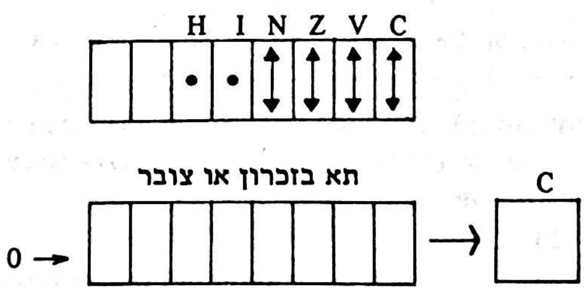
דגל הפסיקה מתואר בסעיף 6.3 . כתוצאה מביצוע ההוראה הדגל מורם ולכן אין אפשרות לבצע את תוכנית הפסיקה, מלבד במקרה של פסיקה בלתי מסיכה (NON MASKABLE INTERRUPT). על הפסיקה נדון בהמשך הפרק, בסעיף 6.10 ובפרק 7.

6.6 הוראות הזזה וסיבוב

הוראות הזזה וסיבוב משנות מיקומן של סיביות במילה.

(1) הזזה לוגית ימינה — Logic Shift Right — LSR

	X	E	
LSR	64	74	
LSRA			44
LSRB			54



כל הסיביות זזות ימינה. הסיבית הפחות משמעותית "נכנסת" לדגל הנשא, והסיפרה אפס נכנסת במקומה של הסיבית המשמעותית ביותר.

ההוראה להזזה ימינה מבצעת חילוק ב-2, כפי שמוצג בדוגמא זו:

```
 1010010
10100100|   10
10
  10
  10
    010
     10
      0
```

התוצאה היא 01010010 ודגל הנשא יורד. תוכל לבדוק זאת על ידי דוגמאות משלך.

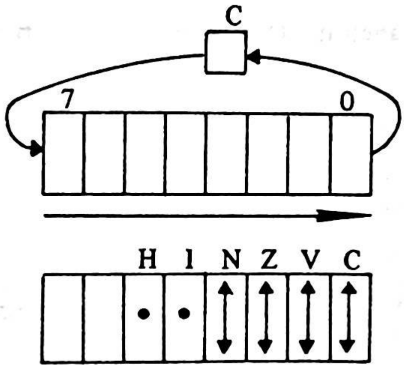
(2) הזזה ימינה אריתמטית   Arithmetic Shift Right   ASR

	X	E	
ASR	67	77	
ASRA			47
ASRB			57

כל הסיביות זזות ימינה, והסיבית הפחות משמעותית "נכנסת" לדגל הנשא. עד כה הפעולה זהה להוראת LSR. השוני הוא שהסיבית המשמעותית ביותר לא משתנה ולכן שתי הסיביות המשמעותיות ביותר זהות.

(3) סובב ימינה   ROtate Right   ROR

	X	E	
ROR	66	76	
RORA			46
RORB			56



עתה הגיע הזמן לאתנחתא קלה לשם תיכנות הוראות של הזזה וסיבוב.

6.7 הוראת הדילוג והוראות ההסתעפות

הוראות ההסתעפות יכולות לשנות את מהלך התוכנית. ההתקדמות לא תהיה אל ההוראה העוקבת אלא אל הוראה אחרת תוך דילוג לפנים או לאחור. בהוראות אלה הדגלים לא משתנים. הוראות אלה שימושיות ליצירת הוראות אלטרנטיביות בהתאם להנחיות התוכנית.

(1) דלג — JMP JuMP

	X	E
JMP	6E	7E

הוראת הדילוג מאלצת מעבר לכל אזור בזכרון לשם המשך ביצוע התוכנית בהתאם לרשום בה וללא כל תנאי. קיימת הוראת דילוג אחת ויחידה. לעומתה קיימות הוראות הסתעפות רבות בהן מצטיין היע"מ 6800. כולן משתמשות במיעון יחסי בלבד ולכן בכולן קיימת מגבלת טווח ההסתעפות בהתאם להיסט.

(2) הסתעף תמיד — BRA BRanch Always

	R
BRA	20

להוראה BRA תפקיד זהה להוראת JMP. השוני טבע מהשימוש במיעון יחסי המגביל טווח הסתעפות עד ל-7F (127_{10}) תאים לפנים ועד 80 (128_{10}) תאים לאחור לכל היותר. ראה סעיף 5.16 הדן במיעון יחסי.

נעיין עתה במספר הוראות "הסתעפות בתנאי". את התנאים להסתעפות מספקים הדגלים שבאוגר מצב.

(3) הסתעף אם דגל האפס מורם — BEQ Branch if EQual zero

	R
BEQ	27

התנאי להסתעפות: Z=1. במילים אחרות, הסתעף אם כתוצאה מביצוע ההוראה הקודמת התקבל אפס.

(4) הסתעף אם דגל הכיוון מונף

BMI Branch if MInus

	R
BMI	2B

תנאי להסתעפות N=1. במילים אחרות, הסתעף אם הפעולה הקודמת גרמה תוצאה שלילית.

כאשר כותבים תוכנית בשפת הסף הכוללת בין השאר הוראות הסתעפות, הכרחי שיהיה אמצעי כלשהו של סימון ההוראות בתוכנית שאליהן מכוונות ההסתעפויות. אמצעי כזה יכולה להיות תוית (LABEL) אשר נכתבת ליד ההוראה שאליה מסתעפת התוכנית. כמובן שצריך לסמן תוית שונה לכל יעד אפשרי של הסתעפות. בזמן תרגום התוכנית לשפת המכונה תסייע התוית לבצע חישובי היסט עבור כל הסתעפות.

כעת הזמן לנסות כתיבת תוכניות עם הוראות הסתעפות.

6.8 הוראות המחסנית

הוראות המחסנית נועדו להעברת מידע בין הצוברים למחסנית ולכן מבחינת הגישה הכללית ניתן לסווגן במסגרת הוראות העברת נתונים. קיימות שתי הוראות מחסנית.

(1) דחוף

PSH PuSH

PSHA	36
PSHA	37

$A \rightarrow M(SP) \quad ; \quad SP-1 \rightarrow SP$

$B \rightarrow M(SP) \quad ; \quad SP-1 \rightarrow SP$

תוכן הצובר עובר לאזור המחסנית בזכרון, אל התא שעליו מורה מצביע המחסנית. לאחר מכן תוכן מצביע המחסנית מופחת ב-1 ובכך הוא מורה על התא הבא הפנוי בה. ראה סעיף 2.7 לתזכורת של פעולת אוגר המחסנית.

(2) משוך

PUL PULL

PULA	32
PULB	33

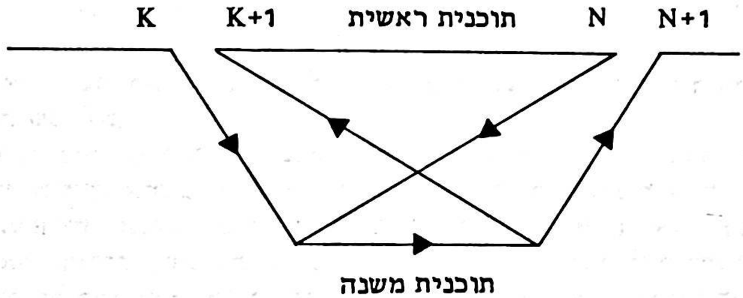
$SP+1 \rightarrow SP \quad ; \quad M(SP) \rightarrow A$

$SP+1 \rightarrow SP \quad ; \quad M(SP) \rightarrow B$

פעולת ה"משיכה" הפוכה לקודמת. שים לב לעובדה שמציין המחסנית מתקדם ב-1 על מנת לחזור לתא התפוס האחרון במחסנית. לאחר מכן התוכן של התא הזה עובר לצובר. מוצע לעיין שוב בסעיף 2.7 אשר דן באוגר מצביע מחסנית.

6.9 הוראות תת-שגרה

הוראות תת שגרה (SUBROUTINE INSTRUCTIONS) גורמות להסתעפות מתוכנית ראשית לתת שגרה הידועה גם בשם תוכנית משנה. ראה איור 6.1:



איור 6.1 – הסתעפות לתוכנית משנה

לפעמים יש צורך לחזור על אותן הפעולות מספר פעמים במהלך הרצת התוכנית הראשית. אפשר לכתוב בכל פעם את כל סדרת ההוראות שגורמות לביצוע של אותן הפעולות, וזו שיטה בזבזנית מדי. לכן השיטה המקובלת היא לכתוב את סדרת ההוראות פעם אחת בלבד, במסגרת **תוכנית משנה** (SUBROUTINE). בכל פעם שזקוקים לתוכנית המשנה אפשר להסתעף אליה מהתוכנית הראשית. לאחר גמר הביצוע חוזרים לתוכנית הראשית וממשיכים אותה מאותה הנקודה שבה הפסקנו קודם לכן.

(1) הסתעף לתוכנית משנה — BSR Branch to SubRoutine

	R
BSR	8D

$PC + 2 \rightarrow PC$

PCL $\overrightarrow{SP}$ STACK

$SP - 1 \rightarrow SP$

PCH $\overrightarrow{SP}$ STACK

$SP - 1 \rightarrow SP$

$PC + OFFSET \rightarrow PC$

(2) דלג לתוכנית משנה — JSR Jump to SubRoutine

	X	E
JSR	AD	BD

$PC + 2 \rightarrow PC$ (FOR x) ; $PC + 3 \rightarrow PC$ (FOR E)

PCL $\overrightarrow{SP}$ STACK

$SP - 1 \rightarrow SP$

PCH $\overrightarrow{SP}$ STACK

$SP - 1 \rightarrow SP$

$ADDRESS \rightarrow PC$

ההבדל בין שתי הוראות אלה הוא בשיטת המיעון.

RTS ReTurn from Subroutine (3) חזור מתוכנית משנה

RTS	39

SP + 1 → SP
PCH $\underrightarrow{SP}$ PC
SP + 1 → SP
PCL $\underrightarrow{SP}$ PC

זוהי ההוראה האחרונה בתוכנית המשנה אשר מחזירה את הפיקוח לרצף ההוראות בתוכנית הראשית. כתובת ההוראה הבאה בתוכנית הראשית שאוחסנה למשמרת במחסנית חוזרת עתה למונה התוכנית (PC). מכאן ואילך תמשיך להתבצע התוכנית הראשית.

6.10 הוראות הפסיקה

מנגנון הפסיקה, בדומה למנגנון תוכנית המשנה, מפסיק זמנית את הביצוע של תוכנית מסויימת על מנת להריץ תוכנית אחרת. אולם מנגנון הפסיקה מורכב בהרבה, כי בנוסף לצורך ל"זכור" היכן הופסקה התוכנית הראשית יש גם צורך לאחסן למשמרת את תוצאות הביניים שאליהן הספיקה התוכנית הראשית להגיע עד לרגע הפסיקה.

נעיין עתה בהוראות הקשורות בפסיקת תוכנה. פסיקות חומרה ומנגנון הפסיקה יוסברו בפרק הבא בסעיפים 7.3.6 עד 7.3.10.

SWI SoftWare Interrupt (1) פסיקת תוכנה

SWI	3F

תגובת המחשב להוראה זו:

א. בדיקת דגל הפסיקה I. אם הוא מונף, המחשב יתעלם מההוראה.

ב. PC + 1 → PC.

ג. אחסון אוגרי היע"מ במחסנית לפי הסדר הבא:

(1) מונה התוכנית - (PC)
(2) אוגר מורה - (IX)
(3) צובר A
(4) צובר B
(5) אוגר מצב - (CCR)

ד. הנפת דגל הפסיקה (כדי לא לאפשר פסיקה לתוך פסיקה).

ה. ביצוע תוכנית הפסיקה החל מהכתובת הנמצאת בתאים FFFA (PCH) ו- FFFB (PCL).

שימוש נפוץ בהוראה זו אפשר לראות בסיום כל תוכנית. משמעות ההוראה SWI בסוף התוכנית היא פסיקת התוכנית בסופה וחזרה לתוכנית ההפעלה (MONITOR) הנמצאת בכל מערכת מחשב ומאוחסנת באזור הזכרון לקריאה בלבד (ROM). יש לציין שניתן לסיים תוכנית גם על ידי הוראות אחרות.

WAI WAIt for Interrupt | (2) המתן לפסיקה

WAI	3E

בתגובה להוראה זו מתבצעות הפעולות לפי סעיפים א, ב, ג של הוראת SWI וכל פעולת המערכת נעצרת. המחשב דרוך במצב הכן לקבלת פסיקת חומרה מהתקן חיצוני ועל כן מאפשר תגובה מיידית לפסיקה אפשרית.

RTI ReTurn from Interrupt | (3) חזור מפסיקה

RTI	3B

זאת הוראת חזרה מתוכנית פסיקה לתוכנית ראשית. סדר ביצוע ההוראה הפוך לסדר הפעולות בהוראה SWI.

6.11 הוראת אי-הפעלה

NOP No OPeration | אל תבצע דבר

NOP	01

PC + 1 → PC

הדבר היחיד שמתבצע הוא הקידום של מונה התוכנית ב-1.
משתמשים בהוראה זו כאשר מעוניינים:

1. להשאיר מקום ריק באיזור התוכנית בזכרון לצורך שימוש אפשרי בשלב מאוחר יותר.
2. להשיג השהייה קצרה.

6.12 חישוב זמן הדרוש לביצוע הוראה

מחשב יבצע הוראה בזמן הנמדד בסדר גודל של מיקרושניות. כל הוראה היא סידרת פעולות קריאה וכתיבה אשר כל אחת מהן נמשכת זמן השווה למחזור השעון (CLOCK CYCLE) או מחזור המכונה (MACHINE CYCLE). עבור מיקרומעבד 6800 שני המושגים האלה חד הם (ראה סעיף 7.2).

להלן נוסחה פשוטה כדי לחשב את הזמן הדרוש לביצוע הוראה:

(מספר מחזורי השעון בהוראה) × (זמן מחזור השעון) = T

זמן מחזור השעון תמצא בספר המלווה מערכת בה אתה משתמש. מספר מחזורי השעון עבור כל הוראה תמצא בנספח א'. מקובל גם לנקוב במספר מחזורי השעון לציון זמן יחסי הדרוש לביצוע הוראות.

6.13 מהי התוכנית היעילה?

התוכנית היעילה היא ללא ספק תוכנית המשיגה את יעדיה מבחינת המשימות בזמן הקצר ביותר ובתפישת זכרון קטנה ביותר. בענין המשימות - נצפה שכולן תושלמנה אך מבחינת הזמן ותפישת הזכרון נוכל לחפש דרך פשרה.

מובן שהדבר הטוב ביותר הוא החסכון בשני הגורמים גם יחד, זמן וגם זכרון. הבעייה היא שלא תמיד ניתן להשיג את שניהם. מבחינת זמן הדרוש לביצוע ההוראה, יידרשו בדרך כלל 2 עד 8 מחזורי שעון. אם נניח מחזור הקצר ביותר האפשרי של 1 מיקרושניה, הרי יידרש זמן בין 2 עד 16 מיקרושניות לביצוע הוראות שונות.

הזמן הדרוש תלוי בשני גורמים:

1. סוג המיעון. במיעון המיידי משיגים את הזמן הקצר ביותר, 2 עד 3 מחזורי שעון. המיעון באמצעות אוגר מורה הוא בדרך כלל הבזבזני ביותר, 5 עד 7 מחזורי שעון, להוציא הוראת BSR בה הוא מגיע ל-8 מחזורי השעון. לשיא של 12 מחזורי השעון מגיע באופן חד-פעמי המיעון המובלע בהוראה של פסיקת התוכנה SWI. הסיבה לזמן הארוך היא הצורך בפעולות קריאה וכתיבה רבות כדי להשלים את פיענוח הכתובות וההוראה.

2. סוג ההוראה. למעשה כמעט בכל סוגי ההוראות ניתן להגיע במיעון מיידי לזמן מינימלי של 2 מחזורי שעון. הוראות ההסתעפות הן הוראות הדורשות זמן ביצוע של 4 מחזורים. ההוראות הארוכות ביותר בזמן ביצוע הן הוראות תת-שיגרה ופסיקה. ושוב השיא שייך להוראה הפופולרית ביותר SWI אשר התופסת תא אחד בלבד אך צורכת 12 מחזורי שעון.

לגבי מספר התאים שתופסת הוראה בזכרון, המצב פשוט יותר. הוראה רשומה בתא אחד, בשניים או בשלושה תאים בהתאם לסוג המיעון בה. להוראה במיעון מובלע דרוש תא זכרון אחד. הוראה במיעון מיידי, ישיר, יחסי ובאמצעות אוגר מורה תירשם בשני תאים. להוראת מיעון מורחב דרושים שלושה תאי זכרון (ראה טבלת סיכום סוגי המיעון בסוף פרק 5).

אולם אין הוראה שיש לה את כל סוגי המיעון, וכן אין סוג מיעון הקיים עבור כל ההוראות.

המסקנה הנובעת מכל האמור היא שצריך לתמרן בין הרצוי לבין המצוי. לעיתים יש לקבוע את סדר העדיפויות, כלומר במה חיוני יותר לחסוך: בזמן או בנפח הזכרון. ההחלטה צריכה להיות לגופו של עניין לגבי כל תוכנית בנפרד. יש לשקול זאת בעת תרגול ובודאי בעת כתיבת תוכניות לביצוע פרוייקטים שונים.

6.14 שאלות חזרה

הערות:

א. העזר נא בנספח א' לפי הצורך

ב. כל תוכנית יש לאחסן בזכרון החל בכתובת 0000

ג. מוצע להוסיף פרושים בעברית בצד מינוח אנגלי בנספח.

1. מצא הוראה ADC –

א. באילו שיטות מיעון ניתן להשתמש בהוראה זו?

ב. מה הם קודי הפעולה המתאימים לשיטות המיעון השונות?

2. מצא הוראה CBA –

א. מדוע לדעתך מסתפקים בשיטת המיעון המובלע בלבד?

ב. כמה מחזורי שעון דרושים לצורך ביצוע ההוראה?

3. מצא הוראה DEC –

א. כמה מילים בהוראה זו עבור המיעון המורחב?

ב. מה פרוש הביטוי M-1→M?

4. מצא הוראה ROLA –

א. מהי מהות פעולתה?

ב. אילו דגלים מושפעים מביצוע ההוראה?

5. מצא הוראה BCC –

א. מה כוונת הביטוי C = 0?

ב. מדוע לדעתך מסתפקים במיעון יחסי בלבד?

6. מצא הוראה NEG –

א. מהו השימוש החסכוני ביותר מבחינת זמן ביצוע ההוראה?

ב. מהו השימוש החסכוני ביותר מבחינת תפוסת הזכרון?

7. מה מבצעת ההוראה שקוד הפעולה שלה הוא:

א. EO

ב. OE

8. השלם את התוכנית הבאה:

0000	86	
0001	26	
0002		LDAB I, 30
0003		
0004		ABA
0005	3F	

9. כתוב תוכנית המאחסנת את המספר 01 בתא 0030 בזכרון:

א. בעזרת הצובר A

פתרון לדוגמא:

הוראה בשפת הסף	הוראה בשפת המכונה	כתובת ההוראה
LDAA I ,01	86	0000
	01	0001
STAA D ,30	97	0002
	30	0003
SWI	3F	0004

ב. ישירות לתא 0030.

10. כתוב תוכנית המחברת מספר 2F עם המספר 50.
תוצאת החיבור תשאר בצובר.

11. בהתייחס לשאלה 9:
א. אילו אוגרים (וצוברים) משתתפים בביצוע התוכנית?
ב. ערוך תרשים המראה השתנות תוכנם של אוגרים אלה תוך ביצוע התוכנית.

12. כתוב תוכנית המכפילה את המספר 3C.
התוצאה תאוחסן בתא 0035 בזכרון.

13. מה מבצעת התוכנית הבאה:

0000	4F
0001	06
0002	3F

14. בכתובות 1000 ו– 1001 נתונים שני מספרים עשרוניים בני שתי ספרות כל אחד בקוד BCD.
כתוב תוכנית החל בכתובת 1010 המכפילה את השניים.
את התוצאה יש לאחסן החל בתא 100A בזכרון.
הדרכה: יש להשתמש בהוראות הזזה וחיבור.

15. נתונה התוכנית הבאה:

```
LDX I, 0123
CLR X,   00
CLR X,   01
CLR X,   02
```

א. מה מבצעת התוכנית?
ב. כמה תאי זכרון תופסת התוכנית?
ג. חשב הזמן הדרוש לביצוע התוכנית, בהנחה שמחזור השעון הינו מיקרושניה אחת.

פרק 7 - חומרת המיקרומחשב עם ה-6800

מטרות ביצועיות לימודיות

עם סיום העיון בפרק זה הקורא יוכל:

1. להסביר את עקרון פעולת שעון המערכת.
2. להסביר את תפקידי כל הקוים המתחברים אל ה–6800
3. להסביר את מנגנון הפסיקה ולהבחין בין כל סוגי הפסיקות
4. להסביר תרשים זרימת פסיקות
5. להסביר תרשים פעולת המיקרומעבד
6. לתאר תרשים מלבנים של זכרון אופייני
7. להסביר את המבנה הפנימי של מישק. תקשורת מקבילית (PIA) תפקידיו, וחיבוריו.
8. להסביר את המבנה הפנימי של מישק תקשורת טורית (ACIA), תפקידיו וחיבוריו.

מונחים בפרק זה

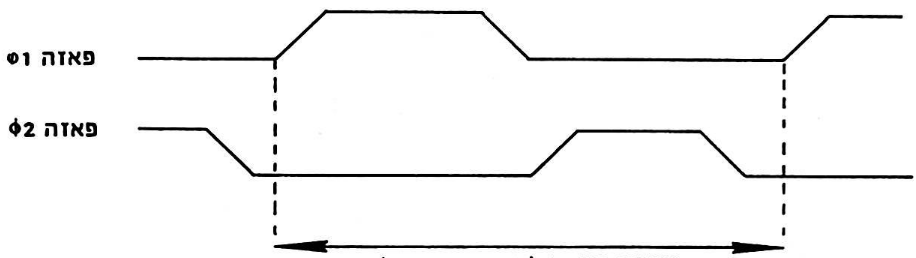
כל מונחי הפרק הנוכחי נמצאים ברשימת הסעיפים.

7.1 מבוא

בפרק 4 הובאה והוסברה מערכת בסיסית עם רכיבי משפחת 6800 באמצעות תרשים מלבנים שבו כל מרכיב הופיע כ"קופסה שחורה". בפרק הנוכחי נדון בכל אחד ממרכיבים אלה באופן מפורט יותר, נפתח את ה"קופסות השחורות" ונכיר את תוכנן. נעזר גם בתרשימים הנמצאים בנספח ב'.

7.2 שעון המערכת

שעון המערכת אשר הוזכר גם בסעיף 2.11 הוא מעגל אלקטרוני המייצר שתי צורות גלים כפי שמוצג באיור 7.1

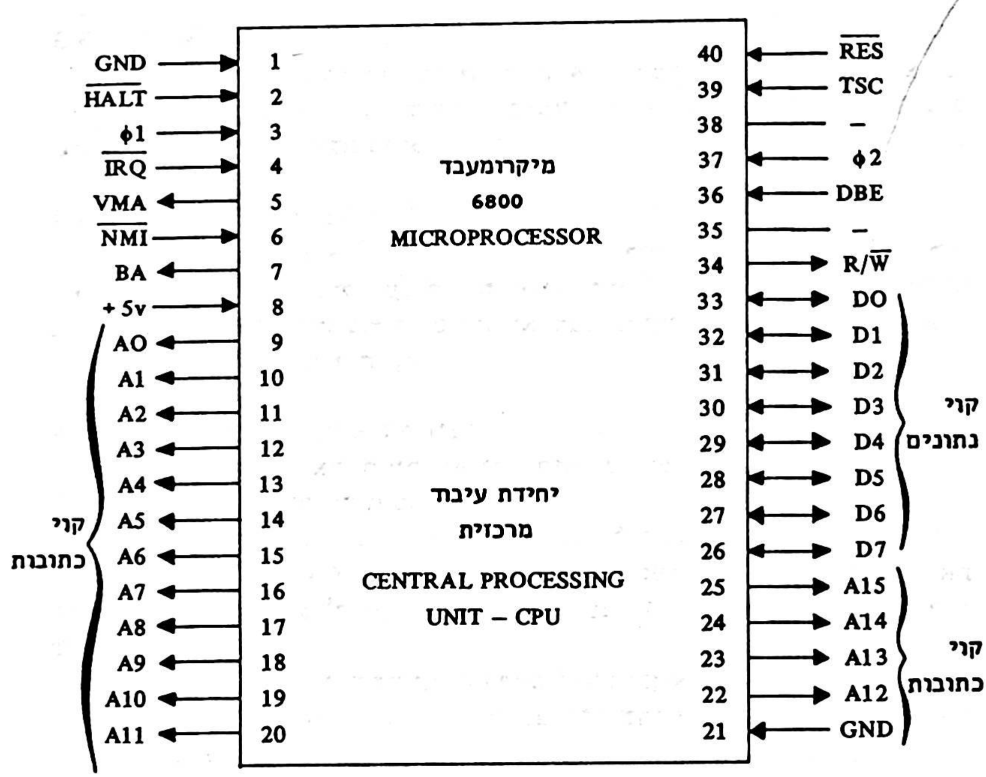


איור 7.1 – שתי הפאזות של מחזור המכונה

7.3 מיקרומעבד 6800

המיקרומעבד מתפקד כיחידת עיבוד מרכזית. הוא בנוי בטכניקה של צפיפות זיווד גבוהה (LSI) ובטכנולוגית NMOS המצטיינת ביכולת שיתוף עם טכנולוגית TTL. צריכת הספק נמוכה ומהירות פעולה סבירה אשר אופיניים לטכנולוגיה של NMOS.

תרשים מלבנים כללי של המיקרומעבד מתואר בסעיף 2.2 ובנספח ב'. נתאר עתה את הפינים השונים של המעגל המשולב לפי איור 7.2 . השווה איור זה עם האיור 2.1 .



איור 7.2: תרשים מפורט של קוי החיבור (פינים) של המיקרומעבד 6800.

קוי הכתובת (פינים 9 עד 20 ו–22 עד 25), קוי הנתונים (פינים 26 עד 33) וקוי ההזנה (פינים 8, 1 ו–22) תוארו בסעיף 2.2 . בהמשך נדון בקוי השעון (פינים 3 ו–37) ובקוי הבקרה (כל שאר הפינים).

7.3.1 כניסות שעון המערכת

תפוקת שעון המערכת, φ1 ו–φ2, נכנסת ליע"מ בפינים 3 ו־37. קצב השעון מכתיב את קצב הפעולה של כל מרכיבי המערכת כפי שנראה בהמשך. זמן המחזור הקצר ביותר האפשרי הוא מיקרושניה אחת.

7.3.2 קו VMA (Valid Memory Address)

קו VMA הוא קו יציאה (מפין 5). באמצעות "1" על קו זה מודיעה היע"מ שהמידע המוצג ברגע זה על קוי הכתובת הוא כתובת אמיתית. כלומר, הצירוף הנמצא על קוי הכתובת אינו מקרי, ויש לפעול לפיו ככתובת (לשם כתיבה בזכרון, למשל).

7.3.3 קו R/W (Read/Write)

קו קריאה/כתיבה הוא קו יציאה (מפין 34). רמה "1" על קו זה מציינת שבזה הרגע מתבצעת פעולת קריאה. באמצעות "0" על קו זה מודיע היע"מ שברגע זה מתבצעת פעולת כתיבה.

7.3.4 קו DBE (Data Bus Enable)

זהו קו כניסה (פין 36) אשר רכיב כלשהו יכול להעבירו למצב "1" כדי לאפשר ליע"מ להציג מילה על קוי הנתונים. המילה המוצגת תכתב ברכיב זה בתנאי שהקו R/W יהיה במצב "0" שהוא מצב כתיבה. קו זה שולט על מעבר הנתונים מהיע"מ אל ציוד היקפי.

7.3.5 תאור פעולת קריאה/כתיבה

כאן נעצור לרגע את תאור הפינים השונים. יש בידינו עתה כלים כדי להבין את פעולות הקריאה והכתיבה.

נבחן תהליך הקריאה של קוד הפעולה של הוראה כלשהי מהזכרון לאוגר ההוראות. (עיין בסעיף 2.10 כתזכורת). נניח כי באוגר מונה התוכנית נמצאת הכתובת של קוד הפעולה וקו R/W נמצא במצב "1" (קריאה). סדר הפעולות של היע"מ הוא:

א. פאזה $\phi1$ של שעון המערכת עולה למצב "1" תוכן אוגר מונה התוכנית מוצג על קוי הכתובת. קו VMA עולה למצב "1" כדי לציין שהכתובת המוצגת הינה כתובת אמיתית.

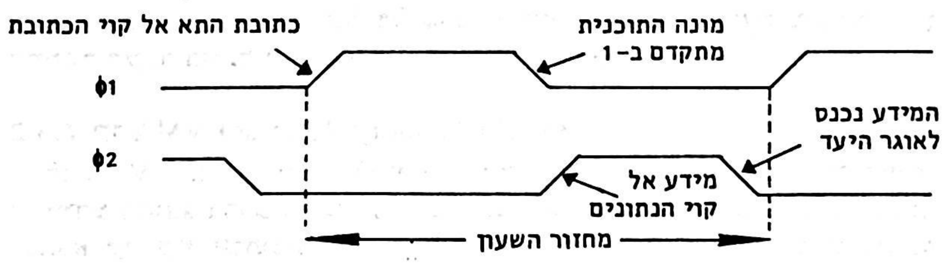
ב. פאזה $\phi1$ יורדת ל-"0". מונה התוכנית מתקדם באחד ($PC + 1 \rightarrow PC$). כעת נמצאת במונה התוכנית כתובת התא הבא בתור.

ג. פאזה $\phi2$ עולה ל-"1". קוד הפעולה הנדרש מוצג על קוי הנתונים.

ד. פאזה $\phi2$ יורדת ל-"0". קוד הפעולה מועבר לאוגר ההוראות (IR) לצורך פיענוח מהות ההוראה על ידי יחידת הבקרה.

כל קריאת מידע כלשהו נעשית בדרך זו ולא רק קריאת קוד הפעולה של ההוראה. ההבדל הוא ביעד שאליו יגיע הנתון. יעדו של מידע שאיננו קוד פעולה הוא אחד מהאוגרים שבשימוש המתכנת (ראה גם סעיף 2.9) ובדרך כלל צובר א' או ב'.

איור 7.3 מסכם את פעולת הקריאה.



איור 7.3: תאור פעולת הקריאה

פעולת הכתיבה מתבצעת באותה הצורה אך ישנם מספר הבדלים בינה לבין פעולת הקריאה:

א) קו $R/\overline{W}$ עובר למצב "0" כדי לציין כתיבה.

ב) קו DBE עובר למצב "1" כדי לאפשר ליע"מ להציג מידע על קוי הנתונים.

7.3.6 מנגנון הפסיקה INTERRUPT

לעיתים, תוך כדי ביצוע תוכנית מסויימת, נוצר צורך להפסיק אותה לצורך ביצוע תוכנית אחרת ולאחר מכן לחזור ולהמשיך את רצף ההוראות. המנגנון המאפשר ומבקר תהליך זה הוא מנגנון הפסיקה.

לפני שיע"מ פונה לתוכנית הפסיקה עליה לאחסן במחסנית (ראה גם סעיף 2.7):

א) את כתובת ההוראה הבאה בתוכנית הראשית שאותה יהיה עליו לבצע לאחר חזרה מתוכנית הפסיקה.

ב) את תוצאות הביניים שהצטברו ביע"מ עד לרגע הפסיקה.

לאחר החזרה מהפסיקה, יחזור כל המידע שאוחסן קודם לכן אל האוגרים בהתאמה והיע"מ תמשיך בביצוע התוכנית הראשית לפי כתובת ההוראה שאוחסנה.

בסעיף 6.10 הוצגה פסיקת תוכנה ועתה נעיין בפסיקת חומרה. ל-6800 יש שתי פסיקות חומרה: פסיקה מסיכה (MASKABLE INTERRUPT) ופסיקה בלתי מסיכה (NON MASKABLE INTERRUPT), אשר תוסברנה בהמשך.

7.3.7 קו $\overline{IRQ}$ (Interrupt Request)

קו הפסיקה הוא קו כניסה (פין 4) אשר מגיע מהתקן חיצוני, דרך מישק תקשורת. במתן "0" על קו זה מבקש ההתקן להריץ עבורו תוכנית פסיקה, אולם זוהי בקשה הניתנת למיסוך על ידי דגל הפסיקה (I) הנמצא באוגר המצב. את דגל הפסיקה ניתן להרים על ידי ההוראה SEI בתוכנית הראשית וכן על ידי תוכנית פסיקה אחרת שמתבצעת ברגע זה.

את דגל הפסיקה ניתן להוריד על ידי הוראת CLI. תגובת היע"מ ל-"0" בקו IRQ דומה מאד לתגובה להוראה SWI, והיא מתבצעת לפי הסדר הבא:

א. בדיקת דגל הפסיקה. אם הוא מונף המחשב מתעלם מבקשת הפסיקה, כלומר יש "מיסוך".

ב. ביצוע ההוראה השוטפת בתוכנית הראשית עד תומה.

ג. קידום מונה התוכנית באחד על מנת שיצביע על כתובת ההוראה הבאה.

ד. אחסון האוגרים במחסנית לפי הנחייתו של אוגר מצביע המחסנית (SP) לפי הסדר הבא:

1. מונה התוכנית (PC)
2. אוגר מורה (IX)
3. צובר א'
4. צובר ב'
5. אוגר מצב (CCR)

סך הכל שבעה תאים רצופים במחסנית בסדר יורד, כלומר כתובת התא בו מאוחסן אוגר המצב היא הכתובת הנמוכה ביותר.

ה. הנפת דגל הפסיקה באוגר המצב, על מנת למנוע פסיקה בתוך פסיקה. אין אפשרות למנוע פסיקה בלתי מסיכה מכיון שאינה תלוייה במצב הדגל.

ו. ביצוע תוכנית הפסיקה החל בכתובת הנמצאת בתאים FFF8 (שמונה הסיביות המשמעותיות יותר, PCH) ו-FFF9 (שמונה הסיביות הפחות משמעותיות, PCL).

כיצד תדע היע"מ על סיום תוכנית הפסיקה? לפי ההוראה האחרונה בתוכנית הפסיקה. הוראה הזו חייבת להיות RTI (חזור מפסיקה לתוכנית ראשית). סדר הפעולות של היע"מ לביצוע ההוראה הינו הפוך לתאור הנ"ל, כלומר הפעולות מתבצעות מסעיף ה' לסעיף א'.

7.3.8 קו $\overline{\text{NMI}}$ (Non Maskable INterrupt)

קו הפסיקה הבלתי מסיכה המגיע מהתקן מבוקר הוא קו כניסה (פין 6). באמצעות "0" על הקו זה ההתקן דורש להריץ עבורו תוכנית פסיקה. הדרישה מתקבלת בכל מקרה, שכן אין חשיבות למצב דגל הפסיקה. תגובת היע"מ מתחילה מסעיף ב' שהוסבר בסעיף 7.3.7, וביצוע תוכנית הפסיקה מתחיל מהכתובת הנמצאת בתאים FFFC (PCH), FFFD (PCL).

תוכנית הפסיקה חייבת להסתיים בהוראה RTI אשר גורמת להמשך הרצתה של התוכנית הראשית, כפי שהוסבר בסעיף 7.3.7 .

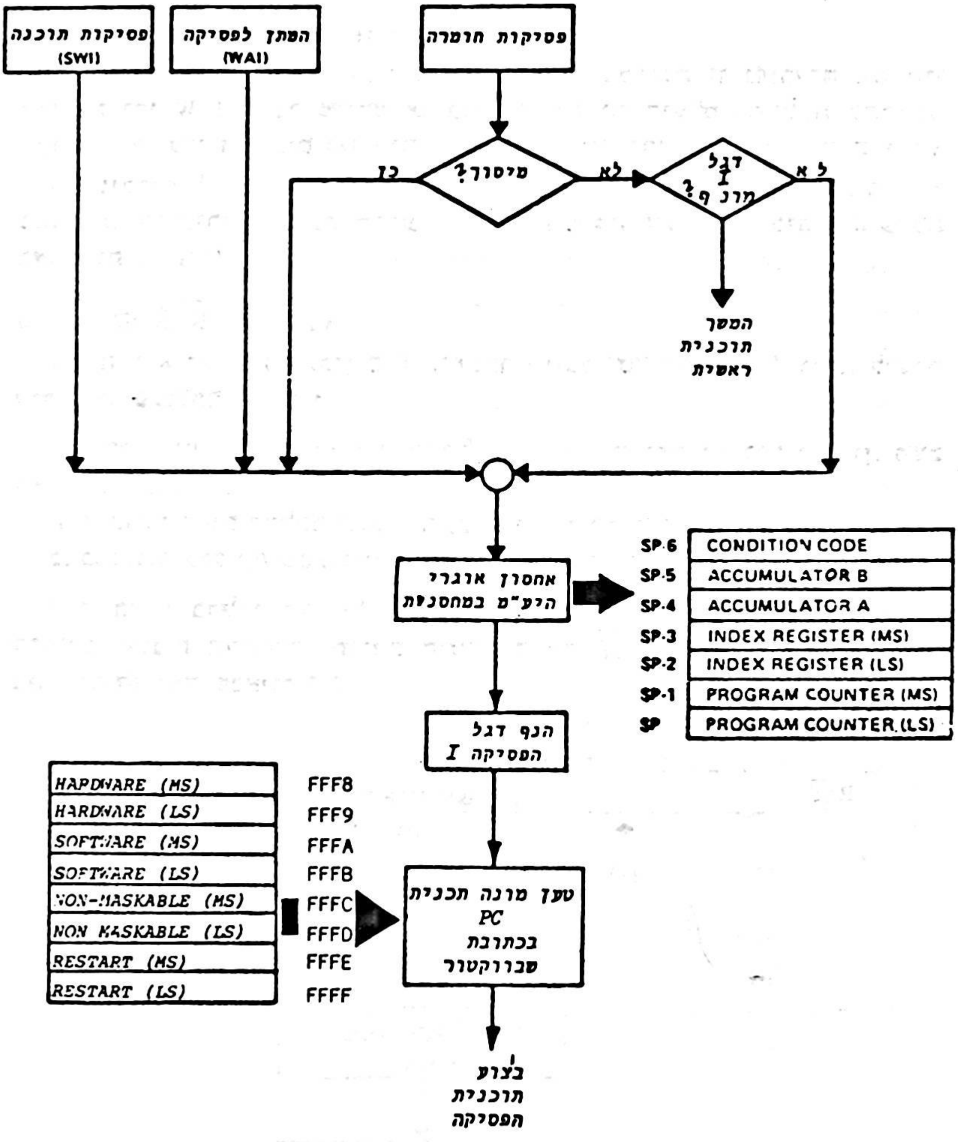
7.3.9 קו $\overline{\text{HALT}}$

קו העצירה הוא קו כניסה (פין 2). כל זמן שהקו נמצא במצבו הרגיל "1" היע"מ תמשיך בפעולותיה. מצב "0" בקו זה יכול להתקבל מהתקן חיצוני והוא יגרום ליע"מ לעצור לאחר שתסיים את ההוראה השוטפת.

כאשר יש "0" בקו העצירה קו ה-VMA עובר למצב "0" וקו BA (יתואר בהמשך) עובר למצב "1". לעומתם כל קוי הכתובות והנתונים קו $R/\overline{W}$ עוברים למצב מנותק (TRI STATE). מצב $\overline{\text{HALT}}$ מנוצל לכמה מטרות, ביניהן האפשרות להריץ תוכנית בצעדים בודדים לצורך בדיקתה ולצורך מעקב אחר ביצועה בשלבים. אם בזמן העצירה תגיע בקשת פסיקת כלשהי היע"מ תטפל בה אבל רק לאחר סיום העצירה.

7.3.10 תרשים זרימה של תהליך הפסיקות

ראה איור 7.4:



איור 7.4 תרשים זרימה של ביצוע פסיקות במיקרומעבד 6800

7.3.11 קו BA (Bus Available)

קו יציאה BA (פין 7) מציין במצבו הרגיל "0" כי קוי הנתונים וקוי הכתובות נמצאים בשליטת היע"מ. במצב "1" היע"מ מודיעה כי קוים אלה הם במצב לוגי מנותק ואינם בשליטתה עוד, ורכיב אחר במערכת יכול לקחת את השליטה עליהם. מצב זה נוצר כתוצאה מאחד התנאים הבאים:

א. עצירת היע"מ על ידי $\overline{HALT}$ כפי שהוסבר בסעיף 7.3.9
ב. על ידי ביצוע ההוראה WAI המתוארת בסעיף 6.10

7.3.12 קו TSC (Tri State Control)

קו TSC הוא קו כניסה (פין 39). רמה "1" בכניסה זו מעבירה את קוי הכתובת וקו $R/\overline{W}$ למצב מנותק או במילים אחרות היע"מ מאולצת להתנתק מקוים אלה. קוי הנתונים נשארים במצב רגיל, מחוברים ליע"מ. הקוים VMA ו־BA עוברים למצב "0" כדי למנוע כתיבה מוטעית מהיע"מ אל רכיב כלשהו במערכת. להזכירך, הוצאת מידע מהיע"מ נשלטת על ידי רכיבי המערכת באמצעות קו DBE.

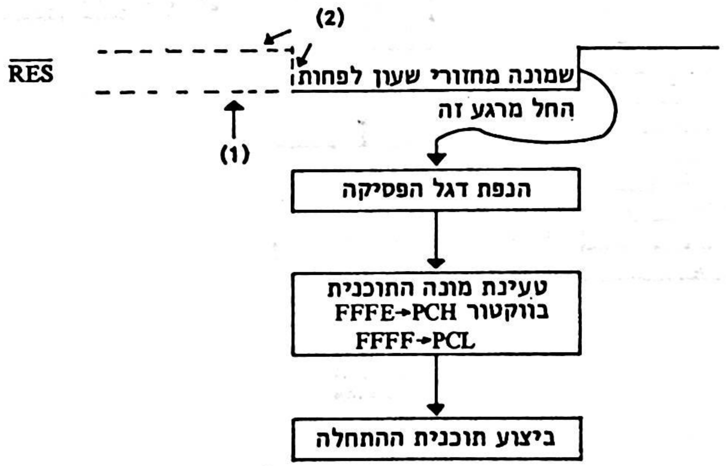
7.3.13 קו $\overline{RES}$ (RESet)

קו זה הוא קו כניסה (פין 40). המעבר ממצב "0" למצב "1" גורם ליע"מ לחדש את פעולתה מהתחלה:

על ידי ביצוע תוכנית הפעלה התחלתית השמורה באופן קבוע בזכרון. מצב ה־ RESET מתהווה:

א. אוטומטית עם הדלקת המערכת (כלומר אספקת מתח הזנה).
ב. כרצוננו, בכל עת שמעוניינים לחזור לתוכנית ההתחלה.

הבט בעיון בחלקו השמאלי העליון של איור 7.5 והבחן בשני המקרים לביצוע תוכנית ההתחלה. תגובת היע"מ למצב $\overline{RESET}$ תהיה לפי המתואר בתרשים הזרימה שבאיור 7.5:



איור 7.5 - תהליך RESET

כל ארבע זוגות התאים בכתובות הגבוהות ביותר בזכרון, מ-FFF8 ועד FFFF, מכילים את הכתובות הראשונות של שלוש תוכניות הפסיקה ושל תוכנית ההתחלה.

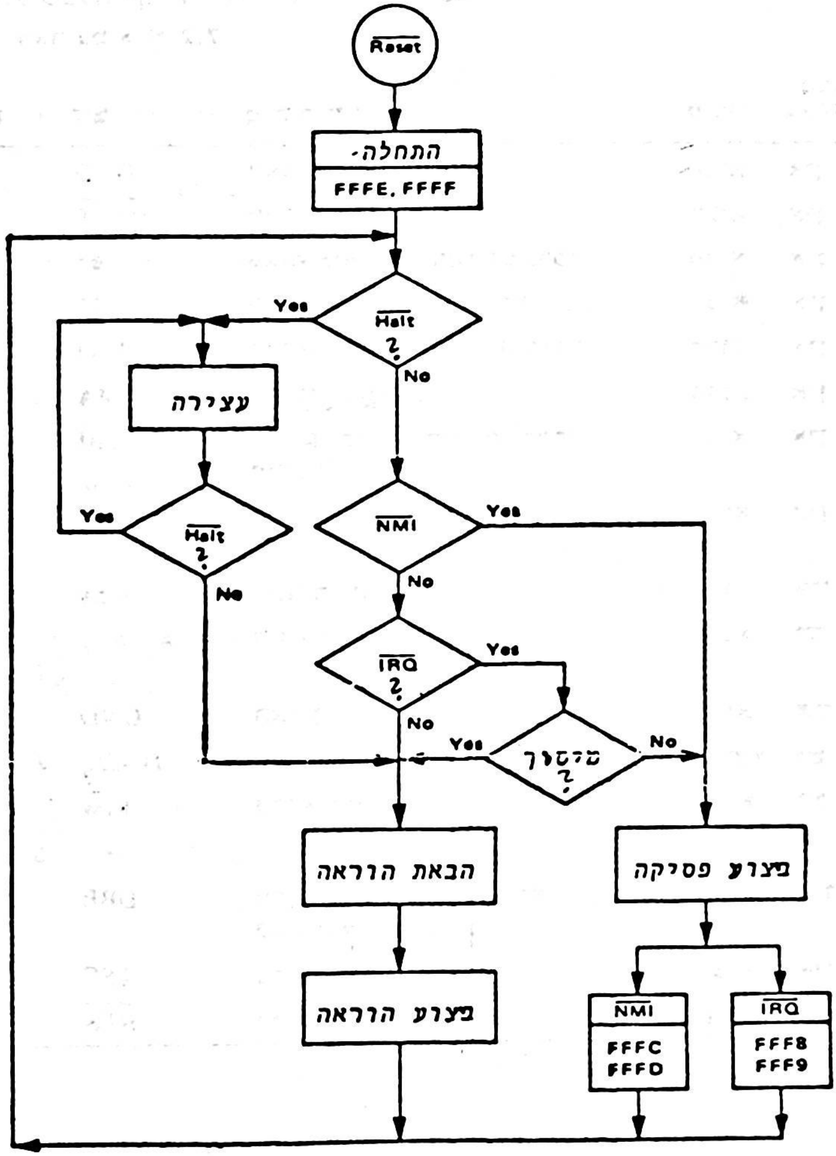
מקובל לכנות כל זוג מתאים בשם ווקטור (VECTOR), משום שהוא מכוון את מונה התוכנית אל התוכנית שעל היע"מ לבצע.

7.3.14 טבלת קוי כניסה/יציאה ב-6800

ראה גם איור 7.2

פין	שם	משמעות	חיבור	מצב מנותק
1	GND	הארקה	מבוא	אין
2	$\overline{HALT}$	עצירת הפעולה	מבוא	אין
3	$\phi 1$	פאזה אחת של שעון המערכת	מבוא	אין
37	$\phi 2$	פאזה שניה של שעון המערכת	מבוא	אין
4	$\overline{IRQ}$	בקשת פסיקת חומרה (מסיכה)	מבוא	אין
5	VMA	כתובת אמיתית	מוצא	אין
6	$\overline{NMI}$	דרישת פסיקת חומרה (בלתי מסיכה)	מבוא	אין
7	BA	קוי כתובת ונתונים לא בשליטת היע"מ	מוצא	אין
8	+ 5v	מתח הזנה	מבוא	אין
9-20 22-25	A_0-A_{15}	קוי הכתובת	מוצא	יש
21	GND	הארקה	מבוא	אין
26-33	D_0-D_7	קוי הנתונים	מבוא/מוצא	יש
34	$R/\overline{W}$	קריאה/$\overline{\text{כתיבה}}$	מוצא	יש
35,38	-	-	-	-
36	DBE	מתן רשות ליע"מ להוציא נתונים לצורך כתיבה	מבוא	אין
39	TSC	בקרת מצב מנותק	מבוא	אין
40	$\overline{RES}$	חזרה לנקודת ההתחלה	מבוא	אין

7.3.15 תרשים זרימה של פעולת המיקרומעבד

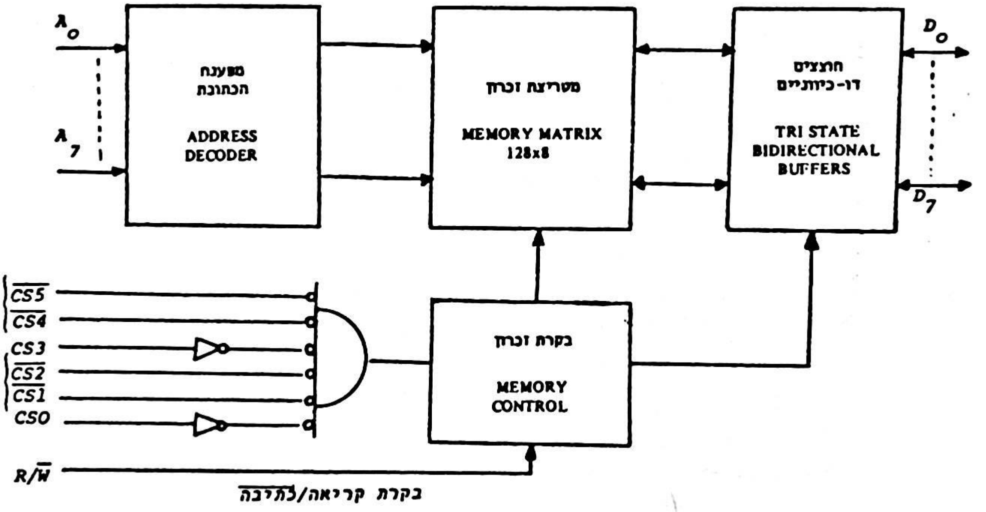


איור 7.6 – תרשים זרימה של פעולת המיקרומעבד 6800

7.4 זכרון לקריאה ולכתיבה, RAM

קיימים מעגלים משולבים רבים ולא רק ממשפחת ה־6800 שניתן להפעיל במערכת כזכרונות. הטכנולוגיה לא חייבת להיות NMOS כמו זו של יע"מ וניתן גם להפעיל זכרון הבנוי בטכנולוגית TTL.

הזכרון שבו משתמשים חייב להיות בעל שלושה מצבים "0" "1" ומצב "מנותק" הן במבוא הנתונים והן במוצא הנתונים. כל מעגל זכרון בנוי כיום במבנה אופייני המתואר באיור 7.7 אשר מציג לדוגמא זכרון לקריאה ולכתיבה RAM, 128x8, MCM 6810.



איור 7.7 – תרשים מלבנים של זכרון אופייני

מקובלים גם זכרונות רבים נוספים כגון 6614↔2114, 6116 ועוד. כאשר אין פנייה של רכיבי המערכת אל המעגל המתואר באיור 7.7, כל החיבורים עוברים למצב מנותק.

7.5 זכרון לקריאה בלבד, EPROM, PROM, ROM

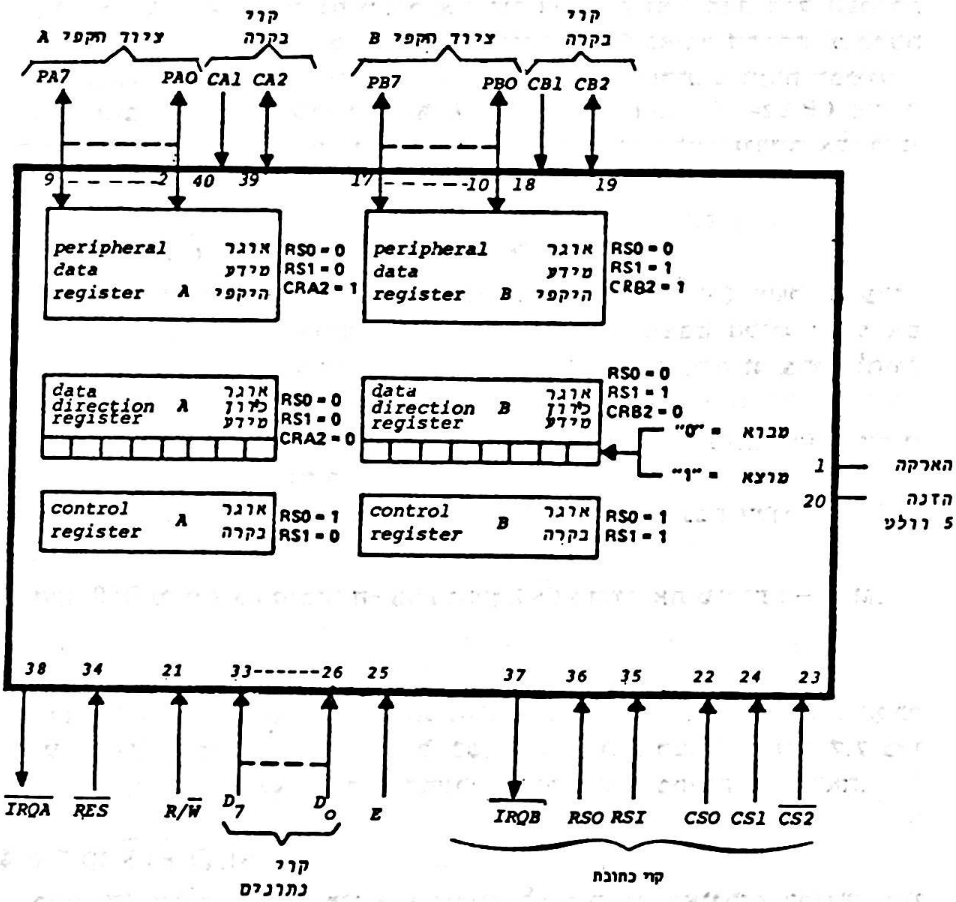
זכרון לקריאה בלבד בנוי במבנה דומה מאוד לזה המתואר באיור 7.7 להוציא קו $R/\overline{W}$ (אשר כמובן מיותר עבור סוג זה של זכרון). בשימוש מקובלים זכרונות קריאה בלבד רבים כגון: 2716, 2732, 6830, 6832 וכו'. נא עיין בנספח ב'.

7.6 מישק תקשורת מקבילית, PIA

מישק תקשורת מקבילית הוא המעגל המשולב MC6821 (אשר מחליף את קודמו MC6820) ובנוי בטכניקת LSI ובטכנולוגית NMOS, ובזיווד של 40 פינים. כדאי לעיין שוב בסעיף 4.2.4 שבו נעשתה ההכרות הראשונית עם מישק התקשורת. המישק כולל:

א. שתי מערכות של קוי נתונים אל שני סוגי ציוד היקפי, כאשר כל קו ניתן למתג בנפרד, על ידי תוכנה כקו יוצא או קו נכנס.
ב. שני אוגרי בקרה הניתנים לתכנות.
ג. שני אוגרי כיוון מידע הניתנים לתכנות.
ד. ארבעה קוי פסיקה המבוקרים על ידי המישק, מהם שני קוים לבקרת ציוד היקפי. זוהי פסיקה המבוקרת על ידי תוכנה.

מישק זה מאפשר התקשורת בצורות שונות ומגוונות כפי שמוצג באיור 7.8 .



איור 7.8 מבנה פנימי של מישק תקשורת מקבילית, PIA

7.6.1 מבנה פנימי

המישק מורכב משתי מערכות זהות A ו– B. עבור כל אחד משני סוגי הציוד המבוקרים יש אוגר מידע היקפי, אוגר כיוון מידע ואוגר בקרה. נעיין לדוגמא במערכת A:

א. באוגר המידע ההיקפי מאוחסן המידע אל הציוד ההיקפי וממנו.

ב. אוגר כיוון המידע ניתן לתכנות על מנת לקבוע את כיוונו של כל אחד מקוי הנתונים, החוצה או פנימה. אם לדוגמא סיבית מספר 3 הינה "1" וסיבית מספר 5 הינה "0" הרי קו מידע 3 "יוצא", וקו מידע 5 "נכנס".

ג. אוגר הבקרה מאפשר ליע"מ לבקר ולתכנת את קוי הבקרה לציוד ההיקפי CA1 ו–CA2. אוגר זה גם מבקר את קוי הפסיקה ואת מצב דגל הפסיקה הפנימי של המישק. בנוסף לכל אלה, סיבית מס' 2 באוגר הבקרה מצטרפת לקוים RS0 ו– RS1 כדי לקבוע לאיזה מהאוגרים הנותרים מיועדת הפנייה.

הבט שוב באיור 7.8 שבו הצירוף RS0 = 0; RS1= 0; CRA2= 1 מהווה כתובת של אוגר מידע היקפי A. רק על ידי צירוף של שלוש סיביות אלו ניתן לפנות לכל אחד משש האוגרים במישק התקשורת.

7.6.2 קוים $CS0$, $CS1$, $\overline{CS2}$ (Chip Select)

אלו הם קוי כניסה (דרך הפינים 22, 24, 23 בהתאמה) אשר מיועדים לבחירת הרכיב ולכן מחוברים לחלק מקוי הכתובת. מצבם הלוגי קובע אם הפנייה של היע"מ היא אל המישק. כדי שהמישק יגיב לפנייה זו, צריך להיות "1" על קו VMA באותו זמן.

הקוים RS0 ו–RS1 גם הם מחוברים לקוי הכתובת. כל חמשת הקוים יחדיו מגדירים כתובתו של האוגר המסוים במישק המסוים.

בסה"כ תופס המישק ארבע כתובות עוקבות בהתאם למצבם הלוגי של RS0 ו–RS1.

הקו RS0 מהווה את סיבית ה– LSB והקו $\overline{CS2}$ מהווה את סיבית ה–MSB.

7.6.3 קו E (Enable)

דרך קו זה (פין 25) מסופק אות הזמון היחיד שמישק התקשורת מקבל מהיע"מ. פין זה מתחבר בדרך כלל ל–$\phi 2$. חזור במידת הצורך לסעיף 7.7 כדי לעיין בתפקיד הפאזה $\phi 2$ של שעון המערכת בעת פעולת כתיבה או קריאה.

7.6.4 קו $\overline{RES}$ (RESET)

רמת "0" על הקו הזה (פין 34) גורמת לאיפוס כל האוגרים במישק. קוי הנתונים אל הציוד ההיקפי יתפקדו כקוי מבוא בלבד וכמוהם הקוים CA2 ו–CB2. כל הפסיקות תבוטלנה.

7.6.5 קו CA1 (CB1) (INTERRUPT INPUT LINE)

על קו כניסה זה דרך פין 40 (ודרך פין 18 בהתאמה עבור ציוד היקפי B) מגיעה בקשת פסיקה מהציוד ההיקפי. כתוצאה מכך מונף דגל הפסיקה שהוא סיבית מס' 7 באוגר הבקרה.

7.6.6 קו CA2 (CB2) (PERIPHERAL CONTROL LINE)

קו זה דרך פין 39 (ודרך פין 19 בהתאמה) ניתן לתכנת כקו יוצא הוא מבקר ציוד היקפי. כקו נכנס הוא מתפקד כקו בקשת פסיקה בנוסף ל-CA1.

הדרך הטובה ביותר ללמוד תכנות היא דרך הניסוי. עתה תוכל לבצע ניסוי בתכנות ה-PIA. קרא בעיון את הרקע התאורטי ופעל בהתאם.

בהצלחה.

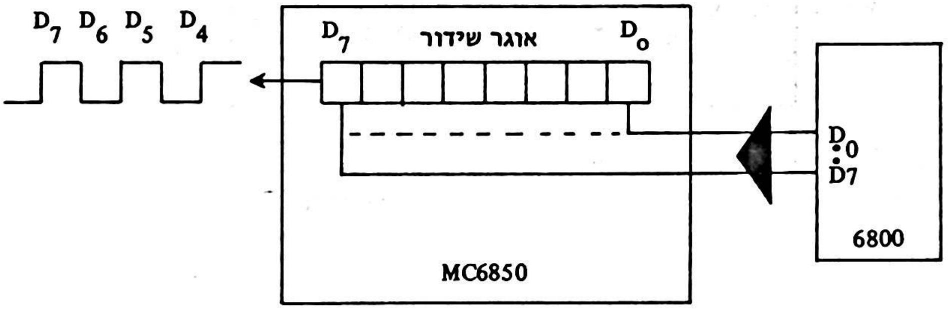
7.7 מישק תקשורת טורית אסינכרונית ACIA

מישק תקשורת טורית הוא המעגל המשולב MC6850. הוא בנוי בטכניקת LSI ובטכנולוגית NMOS בזיווד של 24 פינים. מישק תקשורת טורית ACIA מאפשר קליטה ושידור של סיביות בודדות, בזו אחר זו, באופן אסינכרוני. על ידי כך הוא מתאם בין מחשב אשר פועל באופן מקבילי לבין ציוד היקפי העובד באופן טורי. קצב העברת המידע הטורי הינו עד 500,000 סיביות לשניה, תוך בדיקת הזיוגיות וארך המילה. ההכרות הראשונית עם המישק נעשתה בסעיף 4.2.5, כדאי לעיין בו שוב.

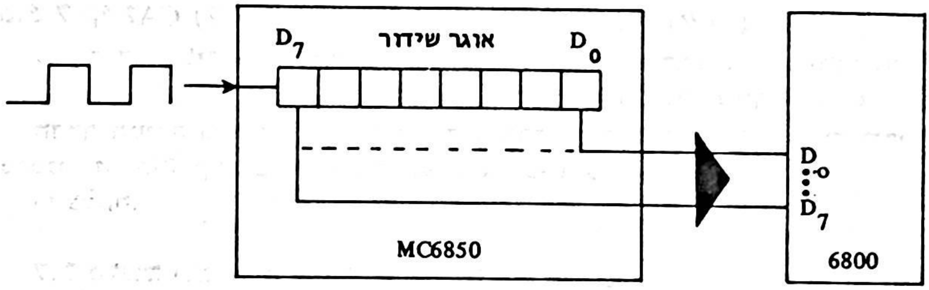
7.7.1 עקרון התקשורת המקבילית- טורית

מישק התקשורת "מתווך" בין היע"מ מחד, לבין הציוד ההיקפי המבוקר מאידך. היע"מ הוא התקן מקבילי והציוד ההיקפי הוא ציוד טורי ועל כן יש צורך במתווך אשר יקיים תקשורת מקבילית עם היע"מ ותקשורת טורית עם הציוד ההיקפי.

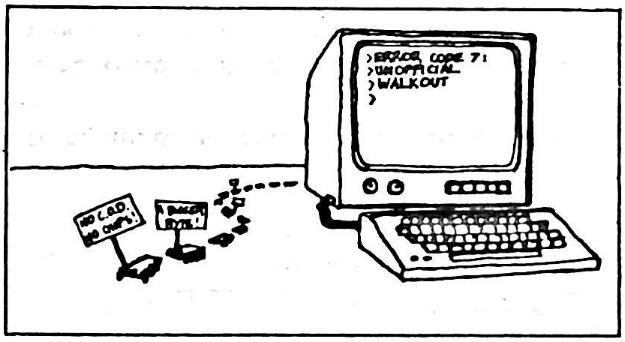
תאור עקרוני של תהליך התקשורת ניתן באיורים 7.9 ו-7.10 .



איור 7.9 - מנגנון המרה מקליטה מקבילית לשידור טורי



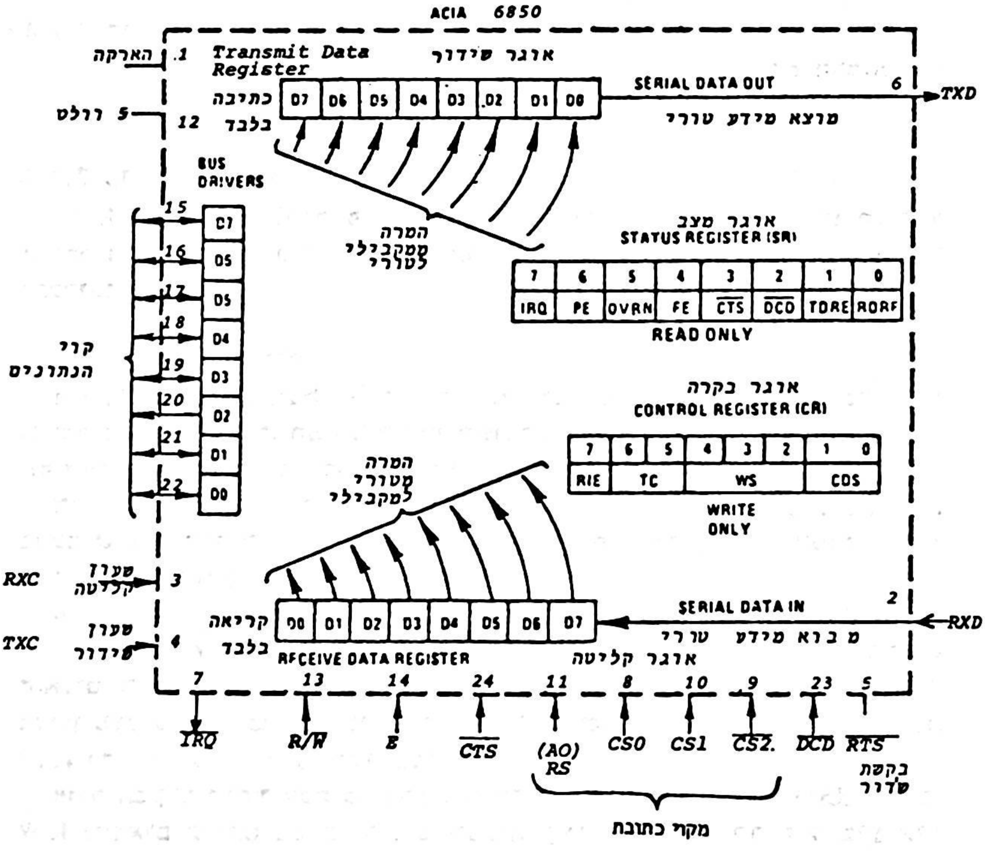
איור 7.10 - מנגנון המרה מקליטה טורית לשידור מקבילי



הודעה: ה"ג'וקים" נוטשים

7.7.2 מבנה פנימי

המבנה הפנימי של המישק מוצג באיור 7.11:



איור 7.11 – מבנה פנימי של מישק תקשורת טורית אסינכרונית, ACIA

7.7.3 קו RS ‏(Register Select)

קו זה הוא כניסה (פין 11) אשר בוררת בעזרת הקו $R/\overline{W}$ זוג אוגרים שידור/קליטה או זוג אוגרים בקרה/מצב. הקו מתחבר לקו הכתובת Ao של היע"מ.

7.7.4 קוים $\overline{CS2}$, CS1, CS0 ‏(Chip Select)

אלה הם קוי כניסה (דרך הפינים 9, 10 ו־8 בהתאמה) לבחירת הרכיב והם מחוברים לחלק מקוי הכתובת.

7.7.5 קו E ‏(Enable)

קו E הוא כניסה (דרך פין 14) שדרכה מגיע אות הזימון מהיע"מ. הוא מחובר בדרך כלל אל פאזה $\phi 2$ של שעון המערכת לצורך זימון פעולת הקריאה והכתיבה.

7.7.6 אוגרים פנימיים

מידע המיועד לכתיבה על ידי היע"מ מומר ממבנה מקבילי למבנה טורי ונשמר באוגר השידור. המידע הטורי משודר אל הציוד ההיקפי דרך הקו TXD (פין 6). קצב השידור מבוקר על ידי שעון שידור חיצוני TXC (פין 4).

המידע מהציוד ההיקפי נקלט דרך קו RXD (פין 2) ומגיע לאוגר הקליטה בקצב שעון הקליטה RXC (פין 3). ברגע מתאים המידע מומר ממבנה טורי למבנה מקבילי ונקרא על ידי היע"מ.

אוגר המצב מיועד לקריאה בלבד. הקריאה מתאפשרת כאשר הקו RS נמצא ב־"0" והקו $R/\overline{W}$ במצב "1". תפקיד האוגר להודיע ליע"מ על המצבים השונים הקשורים בתהליך התקשורת. כל סיבית בו היא למעשה דגל אשר מציין מצב מסויים כגון מוכנות לשידור ($\overline{CTS}$) שיבוש באורך המילה הנקלטת (FE), בדיקת זוגיות (PE), בקשת פסיקה (IRQ) ועוד.

אוגר הבקרה מיועד לכתיבה בלבד. הכתיבה מתאפשרת כאשר קו RS וגם קו $R/\overline{W}$ נמצאים שניהם במצב "0". באמצעות האוגר יכול המתכנת לקבוע איך תתבצע התקשורת: חלוקת יחסי הזמנים לשידור ולקליטה (CDS); אורך המילה (WS); בקרת שידור (TC); תנאי הפסיקה (RIE).

7.8 שאלות חזרה

1. האם יש חשיבות לכך שזמני ה-on וה- off של שעון המערכת אינם שווים? מדוע?
2. תאר בפרוטרוט את מחזור הכתיבה של ה- 6800.
צרף להסברך תרשים בדומה לאיור 7.3
3. הסבר בפרוטרוט את תרשים הזרימה של ביצוע הפסיקות המתואר באיור 7.4
4. כמה תאים במחסנית דרושים כדי לאחסן תוכנו של מונה התוכנית, או של אוגר מורה?
מדוע?
5. תוכן אוגר ההוראות אינו מגיע למחסנית.
מהי הסיבה לדעתך?
6. פרט הבדלים בין מנגנון ה- RESET לבין מנגנון הפסיקה.
7. עיין בתרשים מלבנים של ה- 6800 בנספח ב' ותאר תפקידים של קוי הבקרה.
8. הסבר בפרוטרוט את תרשים הזרימה של פעולות המיקרומעבד המתואר באיור 7.6
9. עיין בדף הנתונים של הרכיב 6830A שבנספח ב'.
א. מה כוונת הסימון 1024×8?
ב. מה כוונת הביטוי Static Operation?
ג. מהו זמן הגישה המירבי לתא כלשהו בזכרון זה?
ד. מה תפקידי הקוים CS?
ה. שרטט מערכת זכרון 8K×8 בעזרת רכיבים אלה.
10. מישק תקשורת מקבילית, PIA
עיין באיור 7.8 ובדפי הנתונים MC 6820 שבנספח ב'.
א. נניח שהכתובת 406C היא נכונה עבור מישק מסויים. התיתכן כתובת 4068 עבור מישק זה? מדוע?
ב. נניח שהכתובת 2BCD הינה הנמוכה ביותר עבור מישק מסויים. כן נניח CRB2=0.
איזהו האוגר המגיב?
ג. סיבית מס' 3 באוגר כוון מידע B הינה במצב לוגי "0". מה פירוש הדבר?
ד. סיבית מס' 7 באוגר הבקרה A הינה במצב לוגי 1. מה פירוש הדבר?

11. מישק תקשורת טורית אסינכרונית, ACIA

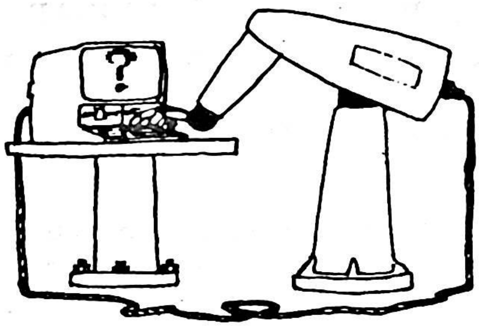
עיין גם בדפי הנתונים MC 6850 שבנספח ב'.

א. פרט בדיקות אמינות המידע על ידי המישק.

ב. מה הם המרכיבים בהם יכולה התוכנה לשלוט במישק זה?

ג. תאר תפקידיו של אוגר מצב.

ד. תאר תפקידיו של אוגר בקרה.



"שיתוף פעולה"

פרק 8 - השוואה בין מיקרומעבדים נפוצים

8.1 מבוא
8.2 מדדי השוואה בין מיקרומעבדים
8.2.1 סוג המיקרומעבד וטכנולוגית הייצור
8.2.2 מתח הזנה, צריכת ההספק ויכולת העמסה
8.2.3 ארכיטקטורה
8.2.4 גודל המילה
8.2.5 מרחב הזכרון
8.2.6 שיטות המיען
8.2.7 אוצר ההוראות
8.2.8 מנגנון הפסיקה
8.2.9 זמן לביצוע ההוראה
8.2.10 אפשרויות התחברות אל סוגי ציוד קלט/פלט
8.2.11 רכיבי תמיכה
8.2.12 מחיר
8.3 מבנה המיקרומעבד 8085 והשוואתו ל־6800
8.3.1 צובר A
8.3.2 זוג האוגרים B–C
8.3.3 זוג האוגרים D–E
8.3.4 זוג האוגרים H–L
8.3.5 אוגר המצב
8.3.6 השוואת תכונות נוספות
8.4 מבנה המיקרומעבד Z־80 והשוואתו ל־6800
8.4.1 המערך היסודי
8.4.2 המערך החילופי
8.4.3 אוגר מורה IX ואוגר מורה IY
8.4.4 אוגר מרענן זכרון
8.4.5 ווקטור הפסיקה
8.5 טבלת השוואה מסכמת
8.6 מבט אל העתיד
8.6.1 מיקרומעבדים
8.6.2 מיקרומחשבים חד־פיסתיים
8.7 שאלות חזרה

מטרות ביצועיות לימודיות:

לאחר סיום העיון בפרק זה יוכל הקורא לערוך השוואה בין המיקרומעבדים 6800, 8085, Z־80 לפי מדדים הנתונים בסעיף 8.2 להבחין בין תכונותיהם.

8.1 מבוא

בפרקים 1, 2, 3 הוצגו נושאים בחומרה ובתוכנה המשותפים למיקרומעבדים רבים. בהמשך, בפרקים 4, 5, 6 ו-7 הוצג המיקרומעבד 6800 בלבד. מטרת פרק זה להציג את שלושת המיקרומעבדים הנפוצים בארץ ולערוך ביניהם השוואה לפי מדדים מקובלים.

6800 מתוצרת מוטורולה (MOTOROLA)
8085 מתוצרת אינטל (INTEL)
Z-80 מתוצרת זילוג (ZILOG)

8.2 מדדי השוואה בין מיקרומעבדים

מדדים (קריטריוניים) עיקריים להשוואת מיקרומעבדים:

א. סוג השימוש וטכנולוגיית הייצור
ב. מתח ההזנה, צריכת ההספק ויכולת העמסה
ג. ארכיטקטורה
ד. גודל המילה
ה. נפח הזכרון
ו. שיטות המיעון
ז. אוצר ההוראות
ח. מנגנון הפסיקה
ט. מהירות הפעולה
י. אפשרות התחברות אל ה"עולם החיצון" - ציוד היקפי
יא. רכיבי תמיכה
יב. מחיר

8.2.1 סוג השימוש וטכנולוגיית הייצור

6800, 8085 ו-Z-80 תוכננו לשימוש כללי (GENERAL PURPOSE) בטכנולוגית NMOS אשר מצטיינת ביכולת זיווד בצפיפות גבוהה, מהירות פעולה טובה, צריכת זרם סבירה ותואמת את משפחת TTL.

8.2.2 מתח ההזנה וצריכת ההספק

שלושת המיקרומעבדים מוזנים על ידי ספק מתח יחיד של 5v. צריכת הזרם הממוצעת היא:

6800	120mA	
8085	170mA	
Z-80	70mA	ובכך יש לנו יתרון בצריכת הזרם הנמוכה יחסית.

כל שלושת המיקרומעבדים מסוגלים להפעיל שער TTL סטנדרטי אחד.

8.2.3 ארכיטקטורה

בארכיטקטורה נדון בהמשך לגבי כל אחד מהמיקרומעבדים בנפרד, בסעיפים 8.3 ו-8.4 .

8.2.4 גודל המילה
לשלושת המיקרומעבדים מילה באורך 8 סיביות בינריות.

8.2.5 מרחב הזכרון
לשלושת המיקרומעבדים 16 קוי כתובת. מכאן שלכולם יש מרחב זהה של 65536 כתובות זכרון (64K). יחד עם זאת, ב- 8085 וב-Z־80 קיימת בנוסף לכך האפשרות לפנות לעד 256 רכיבי קלט/פלט באמצעות קו בקרה I/O שיוסבר בהמשך. ל- 6800 אין קו בקרה כזה.

8.2.6 שיטות מיעון
ככל ששיטות המיעון רבות יותר, כך ניתן לכתוב תוכניות מורכבות וגמישות יותר.

- ל- 6800 קיימות שש שיטות מיעון,
- ל- 8085 קיימות ארבע שיטות מיעון,
- ל- Z־80 קיימות שבע שיטות מיעון, ובכך יש לו יתרון על השניים הקודמים.

8.2.7 אוצר ההוראות
חשוב כמובן שמספר ההוראות יהיה גדול ככל האפשר. אולם גם תחכום ההוראות חשוב לא פחות.

- ל- 6800 יש 72 הוראות בסיסיות ב־ 197 קודי פעולה,
- ל- 8085 יש 80 הוראות בסיסיות ב־ 246 קודי פעולה,
- ל- Z־80 יש 158 הוראות בסיסיות ב־ 692 קודי פעולה.

מכיוון שמילת מחשב אחת יכולה לתאר 256 אפשרויות לכל היותר, יש לעיתים צורך בשתי מילים כדי להביע קוד פעולה של Z־80.

ל- Z־80 יתרון ברור מבחינת אוצר ההוראות.

כדוגמת תחכום נביא את מנגנון הפסיקה של 6800 לעומת זה הקיים ב-8085. בהתקבל תנאי הפסיקה (חומרה או תוכנה), היע"מ מבצעת אוטומטית אחסון של כל האוגרים הפנימיים שלה במחסנית. ב- 8085 יש צורך בהוראת אחסון נפרדת עבור כל אוגר ואוגר והדבר גורם לבזבוז זמן ולבזבוז בנפח הזכרון.

8.2.8 מנגנון הפסיקה
מנגנון הפסיקה העדיף הוא מנגנון שבו קיימות אפשרויות רבות יותר של פסיקות חומרה בדרגות עדיפות שונות. הצורך בהעדפה נוצר כאשר מספר רכיבים מבקשים פסיקה בעת ובעונה אחת. גם לפסיקות תוכנה נודעת חשיבות.

- ל- 6800 שתי אפשרויות פסיקת חומרה, $\overline{NMI}$ ו- $\overline{IRQ}$,
- ל- Z־80 הפסיקות $\overline{NMI}$ ו־$\overline{IRQ}$ זהות לאלה של 6800 בהתאמה,
- ל- 8085 חמש אפשרויות פסיקה, מהן TRAP ו-INTR, אשר זהות לשתים

של Z-80 ו־6800 בהתאמה. ועוד שלוש פסיקות יחודיות RST 7.5, 6.5 RST ו־5.5 RST לפי סדר עדיפות יורד. פרט לזאת, לבקשת פסיקת INTR ניתן למתג עד שמונה תוכניות פסיקה שונות המאוחסנות בתאי הזכרון לקריאה בלבד. לשמונה תוכניות משנה אלה, רמת עדיפות זהה.
ל- 8085 יתרון ברור מבחינת מנגנון הפסיקה.

8.2.9 זמן לבצוע הוראה

בטבלה הבאה ניתנים מחזור שעון מינימלי לעומת זמן מינימלי לבצוע הוראה, עבור שלושת סוגי המיקרומעבדים.

סוג המיקרומעבד	מחזור שעון מינימלי µsec	זמן מינימלי לביצוע הוראה µsec
8085	0.33	1.3
Z-80	0.4	1.6
6800	1	2

למרות שההשואה בין זמני השעון המרכזיים היא חד-משמעית, הרי קשה להקיש מכך על זמני ביצוע הוראות בפועל. הוראה זהה עשוייה להתבצע בתוך כמות שונה של מחזורי שעון במעבדים מסוגים שונים. לדוגמא: הוראת החיבור עם נשא ADC תבוצע ב-6800 בתוך שני מחזורי שעון, כלומר 2 µsec וב־8085 - בתוך שבעה מחזורי שעון, כלומר 2.3 µsec .

מסקנה: אין אפשרות לקבוע איזהו המיקרומעבד המהיר. ניתן רק להשוות את זמני הביצוע של תוכנית נתונה.

8.2.10 אפשרויות התחברות אל סוגי ציוד קלט/פלט

ככל שיש אפשרויות רבות יותר להתחברות מקבילית וטורית, כך ניתן להרבות בסוגי ציוד קלט/פלט. לכל סוגי המיקרומעבדים קיימת אפשרות לתקשורת מקבילית באמצעות קוי הנתונים.

8085 כולל בנוסף שני קוים לתקשורת טורית ישירות מהיע"מ, קו SID לקליטה וקו SOD לשידור. שאר המיקרומעבדים לא כוללים קוים כאלה ואת התקשורת הטורית ניתן לבצע בהם באמצעות מישקי תקשורת טורית.

ב־6800 ניתן לבצע תקשורת טורית (למשל, לרשם קול או למדפסת טורית) גם באמצעות מישק תקשורת מקבילית (PIA). מישק זה קיים במערכת מינימלית (ראה סעיף 4.2) והוא ניתן לתכנות למטרה זו.

8.2.11 רכיבי תמיכה

המיקרומעבד מהווה את מרכז המערכת, אולם הוא לבדו חסר תועלת ללא שאר רכיבי המערכת. כל חברה המייצרת את המיקרומעבד מייצרת גם רכיבים נוספים הנקראים רכיבי תמיכה והנדרשים לצורך הרכבת המערכת כולה.

כל מערכת חייבת לכלול בנוסף ליע"מ, שהוא המיקרומעבד, את הרכיבים הבאים:

א. זכרון לקריאה בלבד
ב. זכרון לקריאה ולכתיבה
ג. מעגלי קלט/פלט, כגון מקשים ותצוגה בהתאמה

בנוסף לכך יש לחבר ל-6800 ול-Z-80 מעגל מתנד עבור השעון המרכזי. ב-8085 המתנד הוא פנימי, אך יש צורך להוסיף גביש חיצוני.

המיקרומעבד העדיף הוא זה הנהנה מרכיבי תמיכה רבים. לשלושת המיקרומעבדים הנדונים קיימת מערכת רכיבי תמיכה ענפה, הכוללת זכרונות, מישקים, בקרים למטרות מוגדרות ועוד.

8.2.12 מחיר

המחיר הוא אחד המדדים החשובים. בתכנון מערכת יש להביא בחשבון את הגורמים הבאים :

א. מה מקבלים תמורת המחיר שמשלמים (אילו רכיבים)?
ב. מהן האפשרויות לשכלול ולהרחבת המערכת בעתיד?
ג. מיהם היצרן והיבואן העומדים מאחורי המוצר מבחינה מקצועית וארגונית, מבחינת שיווק, חלקי חילוף וכו'?
ד. שיקולים ענייניים לגבי מטרת המערכת הספציפית.

הערה: יש לציין, כי המחירים נתונים לשינויים תכופים ולכן את ההשואה מבחינת המחיר יש לבצע סמוך למועד הרכישה.

8.3 מבנה המיקרומעבד 8085 והשוואתו ל-6800

מיקרומעבד 8085 מתוצרת INTEL מהווה שכלול של מיקרומעבד שקדם לו (8080), אף הוא מתוצרת INTEL. בנוסף להשוואות הכוללות שנערכו בסעיפים הקודמים נשווה עתה בקצרה את האוגרים של היע"מ אשר נמצאים בשימוש המתכנת. ראה איור 8.1 .

0	אוגר מצב F						7
CY		P		AC		Z	S

7 — 0	7 — 0
ACCUMULATOR A צובר	(flags F)
REGISTER B אוגר	REGISTER C אוגר
REGISTER D אוגר	REGISTER E אוגר
REGISTER H אוגר	REGISTER L אוגר

15		0
STACK POINTER	S.P.	מצביע מחסנית
PROGRAM COUNTER	P.C.	מונה תוכנית

איור 8.1 - מבנה פנימי של מיקרומעבד 8085

8.3.1 צובר A

תפקידים של צובר A זהים בשתי המעבדים . ב–8085 קיים צובר אחד בלבד, אך במידת הצורך ניתן להעזר בכל אחד מהאוגרים H, E, D, C, B ו–L. כאשר רוצים להשתמש במידע באורך של שש עשרה סיבינות ניתן לעבוד עם זוגות האוגרים B–C, D–E, H–L .

8.3.2 זוג האוגרים B–C

כל אוגר הוא עצמאי ויכול לתפקד כאוגר ל–8 סיביות. אולם ניתן להפעילם גם כזוג אוגרים צמודים. במקרה כזה הם משמשים כאוגר ל–16 סיביות עבור אגירת כתובות או נתונים. המילה המשמעותית יותר נטענת באוגר B, והמילה הפחות משמעותית באוגר C.

8.3.3 זוג האוגרים D–E

תפקידם ואופן פעולתם זהים לאלו של זוג האוגרים B-C. הם יכולים לתפקד בנפרד כאוגרים לשמונה סיביות או כזוג האוגרים 16 סיביות. המילה בעלת המשמעות הגבוהה יותר נמצאת באוגר D.

8.3.4 זוג האוגרים H–L

תפקידם ואופן פעולתם כמו בשני זוגות האוגרים האחרים. מקובל לנצלם לתפקיד נוסף כמו התפקיד של אוגר מורה (IX) ב–6800. ניתן להשתמש בתוכן הטעון בזוג אוגרים זה ככתובת של נתון אבל אין אפשרות להוסיף היסט כלשהו.

בצורה דומה ניתן להפעיל את זוגות האוגרים B-C, D-E. המיעון המשתמש בזוג אוגרים ככתובת של נתון נקרא מיעון עקיף על ידי אוגרים. מיעון בעזרת אוגר מורה כפי שקיים ב–6800 משוכלל יותר בגלל האפשרות של היסט.

8.3.5 אוגר מצב

אוגר המצב ב–8085 נקרא אוגר דגלים (FLAGS). לאוגר חמישה דגלים אשר (ראה איור 8.1) כמעט כולם זהים לאלו של ה–6800.

א. דגל הנשא – CY

ב. דגל הזוגיות – P (PARITY)

דגל זה בודק את מספר הסיביות הנמצאות במצב "1" כתוצאה מביצוע הוראה כלשהי. דגל הזוגיות מורם מסמל מספר זוגי של "1"–ים.

ג. דגל הנשא למחצה – AC

ד. דגל הסימן – S

8.3.6 השואת תכונות נוספות

א. 8085 מנוצל בעיקר בתעשיה, בעוד ש–6800 נוח גם ללימוד.

ב. ל–8085 מנגנון פסיקה מסורבל מבחינת התוכנה הכרוכה בכך. לעומתו מנגנון הפסיקה של 6800 הינו תהליך אוטומטי שאינו זקוק להתערבות

התוכנה. כנ"ל לגבי מנגנון התת-שגרה.

ג. ל-8085 יותר קוי בקרה ופסיקה במספר פינים זהה (40) ולכן יש צורך בריבוב (MULTIPLEXING) של קוי הנתונים עם חלק מקוי הכתובת. כלומר, יש שמונה קוים המשמשים בתפקיד כפול ונקראים DA7-DA0. ב-6800, לכל קו יש תפקיד מוגדר ואין ריבוב תפקידים.

ד. ב-6800 קיימות הוראות הסתעפות רבות יותר מאשר ב-8085.

ה. הדגלים ב-8085 מגיבים רק להוראות אריתמטיות ולוגיות המבוצעות על הצובר בלבד. ב-6800 מושפעים הדגלים גם מפעילויות על הזכרון.

ו. סדר כתיבת הכתובת בתוך הוראה הוא כזה:

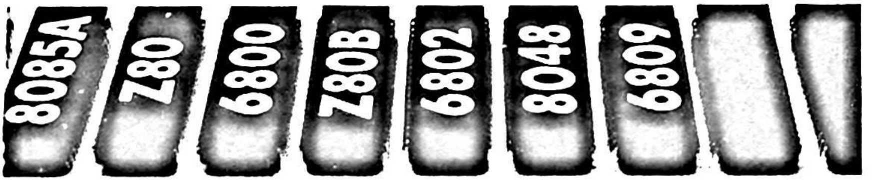
ב-6800 נרשמת קודם המילה בעלת המשמעות הגבוהה יותר כפי שאדם נוהג לכתוב בחיי היום היום,

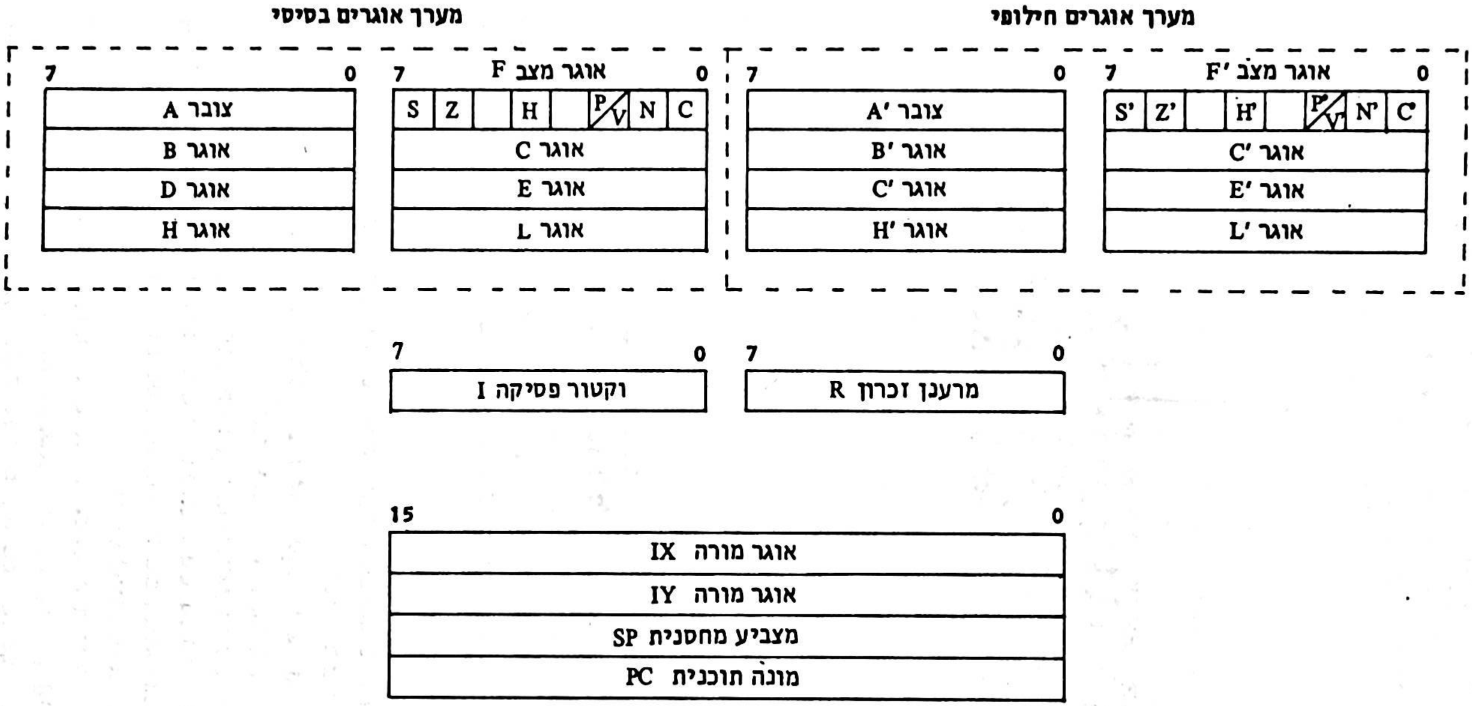
ב-8085 נרשמת קודם המילה בעלת המשמעות הנמוכה יותר.

8.4 מבנה המיקרומעבד Z-80

Z-80 מהווה פיתוח על בסיס 8080, ולכן הוא דומה מבחינות רבות ל-8085, אך הוא משוכלל ממנו מבחינות רבות. יש לו כמעט את כל ההוראות של 8085 (למעט SID ו-SOD) ועוד 80 הוראות נוספות.

ל-Z-80 אוגרים רבים הכוללים גם מערך חילופי בתוספת למערך הבסיסי. מבנה היע"מ והאוגרים שבשימוש המתכנת ניתן באיור 8.2, ומוסברים בהמשך.





איור 8.2 - מבנה פנימי של מיקרומעבד Z-80

8.4.1 המערך היסודי

המערך היסודי זהה לזה של 8085 שהוסבר קודם, פרט לתוספות הבאות באוגר מצב:

א. דגל החיסור (SUBTRACT FLAG ;N)

כאשר דגל זה מונף הוא מסמל שהתוצאה תוצג במבנה עשרוני לאחר פעולת חיסור של מספרים בצופן BCD.

ב. דגל ההצפה/זוגיות (PARITY/OVERFLOW; P/V)

לדגל זה שני תפקידים בהתאם לסוג ההוראה המתבצעת.

1. עבור פעולה אריתמטית על מספרים מכוונים הוא משמש דגל הצפה כמו ב-6800;
2. עבור פעולה לוגית, הזזה והוראות קלט הוא מתנהג כדגל זוגיות.

8.4.2 המערך החילופי

המערך החילופי מבטל חלקית את תפקיד המחסנית. אפשר לאחסן בו את תוכן האוגרים במקרה של פסיקה או תת-שגרה.

8.4.3 אוגר מורה IX ואוגר מורה IY

לאוגרים אלו יש את היתרונות הבאים לעומת האוגר הבודד IX ב־6800:

א. אפשר לטעון בהם שתי כתובות מאזורים שונים בזכרון ועל ידי כך לשלוט על נפח כפול באמצעות ההיסט.

ב. את ההיסט ניתן לתת כמספר מכוון לשני הכיוונים בדומה למיעון יחסי ב-6800 כלומר עד 127_{10} תאים קדימה ועד 128_{10} תאים אחורנית, ולא כמו ב-6800 בו אפשר להוסיף עד 255 תאים קדימה בלבד.

8.4.4 אוגר מרענן זכרון

מאפשר רענון אוטומטי של זכרון לקריאה ולכתיבה. הרענון מתבצע בשלב הבאת קוד הפעולה (FETCHING) של כל הוראה. בכך מתאפשר השימוש בזכרון דינמי המהיר יותר מזכרון סטטי.

8.4.5 וקטור פסיקה (INTERRUPT VECTOR; I)

וקטור הפסיקה הוא אוגר אשר מאחסן את המילה המשמעותית יותר של כתובת ההוראה הראשונה בתוכנית הפסיקה. על ידי כך נוצרת גישה ישירה לתוכנית הפסיקה מבלי שיהיה צורך לעבור דרך כתובת מכוונת, כפי שזה נעשה ב-6800. על ידי כך חוסכים זמן.

8.5 טבלת השואה מסכמת

הערך המושווה	6800	8085	Z-80
צריכת זרם ממוצעת (mA)	120	170	70
מתח הזנה	5V	5V	5V
כמות הפינים	40	40	40
גישה ישירה לזכרון (DMA)	כן	כן	כן
מספר אפשרויות פסיקה	3	6	3
שעון בתוך היע"מ ?	לא	כן	לא
זמן מירבי להוראה (USEC)	5	5.85	8
זמן מזערי להוראה (USEC)	2	1.3	1.6
מספר פאזות השעון	2	2	2
תדר שעון מירבי (MHz)	1	3.3	2.2
מספר קודי פעולה	197	246	692
מספר הוראות בסיסיות	72	80	158
מרחב זכרון ישיר	64K	64K	64K
סיביות במילה	8	8	8
התאמה ל- TTL	כן	כן	כן
טכנולוגיה	NMOS	NMOS	NMOS

8.6 מבט אל העתיד

פיתוח נושא המיחשוב מתבצע כיום בעולם בשני מישורים:

א. מיקרומעבדים

ב. מיקרומחשבים חד-פיסתיים

בעתיד נראה שיכלול הציוד, איכות גבוהה יותר, גמישות בהרכבת מערכת, גמישות בתיכנות ומהירות פעולה גבוהה יותר.

8.6.1 מיקרומעבד

פיתוח הנושא צועד לקראת המטרות הבאות:

א. נפח זכרון ומהירות גישה אליו: 64M וזמן גישה (ACCESS TIME) בסדר גודל של עשרות ננושניות. (ננשניה - אלפית מיקרושניה. 10^{-9} של שניה.)

ב. גודל מילה: 16 וגם 32 סיביות.

ג. אוצר הוראות ושיטות מיען: פיתוח מנגנון תוכנה מתוחכם כדי שהוראה אחת תחליף סדרת הוראות הידועות היום. לדוגמא: הוראת כפל או חילוק.

ד. מהירות פעולה: פיתוח טכנולוגיות חדישות שתאפשרנה לעבוד בתדר שעון מרכזי של 14MHz.

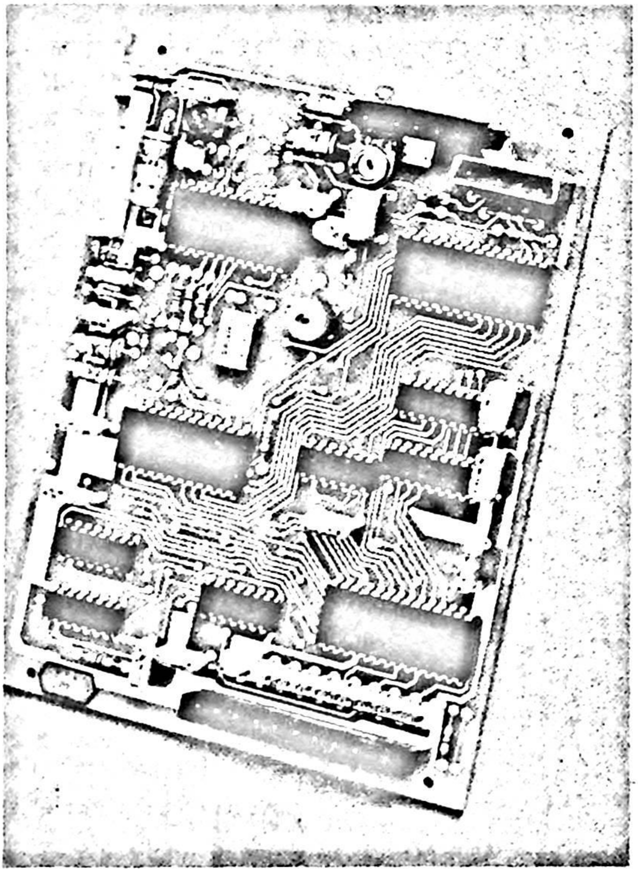
8.6.2 מיקרומחשב חד־פיסתי

קיימים כיום מיקרומחשבים שלימים אשר מזוודים במעגל משולב אחד. לדוגמא: MOTOROLA 6870, INTEL 8748, ZILOG Z-8. פיתוח הנושא צועד לקראת המטרות הבאות:

א. מרחב הזכרון: 8K זכרון לקריאה בלבד (ROM) ו־0.5K זכרון לקריאה וכתיבה (RAM).

ב. גודל המילה: 16 סיביות

ג. מרכיבי המערכת: (באותו המעגל המשולב עצמו יותקנו מישק תקשורת מקבילית ומישק תקשורת טורית ברי תיכנות, מונים, מעגלי זימון ברי תיכנות, ממיר אות תקבילי לספרתי (ANALOG/DIGITAL; A/D) וממיר אות ספרתי לתקבילי (DIGITAL/ANALOG; D/A).



"ג'וקים"

8.7 שאלות חזרה

1. ערוך טבלה המפרטת את המיקרומעבד המושלם ביותר עבור כל אחד מהמדדים מסעיף 8.2 .

2. ערוך טבלה המפרטת את המיקרומעבד המושלם ביותר עבור כל אחד מהגורמים מסעיף 8.5 .

3. ערוך השוואה בין מיקרומעבדים מסוג 8085 ו–6800 מבחינת הצוברים והאוגרים.

4. ערוך השוואה בין מיקרומעבדים מסוג 8085 ו– 6800 מבחינת מנגנון הפסיקה:
   א. בחומרה
   ב. בתוכנה.

5. ערוך השוואה בין מיקרומעבדים מסוג Z-80 ו– 6800 מבחינת הצוברים והאוגרים.

6. ערוך השוואה בין מיקרומעבדים מסוג Z-80 ו– 6800 מהבחינות הבאות:
   א. גמישות בתוכנה
   ב. זמן לביצוע הוראה ביחס לזמן המחזור של השעון המרכזי.

7. סקור מגמות הפיתוח העתידי של:
   א. מיקרומעבדים
   ב. מחשבים חד–פיסתיים.

נספח א' - תוכנת מערכת עם ה-6800

נספח זה כולל טבלאות עזר להכרת ההוראות של המיקרומעבד.

א-1 : הוראות ה-6800 לפי כינוי ההוראה.

א-2 : טבלאות ריכוז הוראות

(בציון קוד פעולה לפי שיטות מיעון שונות, מהות הפעולה, אוגר מצב).

- הוראות הצוברים והזכרון
- הוראות דילוג והסתעפות
- הוראות אוגר מורה ומחסנית
- הוראות אוגר מצב

א-3 : הוראות ה-6800 לפי קוד פעולה הקסדצימלי.

א-4 : מספר מחזורי שעון הדרושים לביצוע הוראות.

הוראות ה-6800 לפי כינוי הוראה

MICROPROCESSOR INSTRUCTION SET – ALPHABETIC SEQUENCE

ABA	Add Accumulators
ADC	Add with carry
ADD	Add
AND	Logical And
ASL	Arithmetic Shift Left
ASR	Arithmetic Shift Right
BCC	Branch if Carry Clear
BCS	Branch if Carry Set
BEQ	Branch if Equal to Zero
BGE	Branch if Greater or Equal Zero
BGT	Branch if Greater than Zero
BHI	Branch if Higher
BIT	Bit Test
BLE	Branch if Less or Equal
BLS	Branch if Lower or Same
BLT	Branch if Less than Zero
BMI	Branch if Minus
BNE	Branch if Not Equal to Zero
BPL	Branch if Plus
BRA	Branch Always
BSR	Branch to Subroutine
BVC	Branch if Overflow Clear
BVS	Branch if Overflow Set
CBA	Compare Accumulators
CLC	Clear Carry
CLI	Clear Interrupt Mask
CLR	Clear
CLV	Clear Overflow
CMP	Compare
COM	Complement
CPX	Compare Index Register
DAA	Decimal Adjust
DEC	Decrement
DES	Decrement Stack Pointer
DEX	Decrement Index Register
EOR	Exclusive OR
INC	Increment
INS	Increment Stack Pointer
INX	Increment Index Register
JMP	Jump
JSR	Jump to Subroutine
LDA	Load Accumulator
LDS	Load Stack Pointer
LDX	Load Index Register
LSR	Logical Shift Right
NEG	Negate
NOP	No Operation
ORA	Inclusive OR Accumulator
PSH	Push Data
PUL	Pull Data
ROL	Rotate Left
ROR	Rotate Right
RTI	Return from Interrupt
RTS	Return from Subroutine
SBA	Subtract Accumulators
SBC	Subtract with Carry
SEC	Set Carry
SEI	Set Interrupt Mask
SEV	Set Overflow
STA	Store Accumulator
STS	Store Stack Register
STX	Store Index Register
SUB	Subtract
SWI	Software Interrupt
TAB	Transfer Accumulators
TAP	Transfer Accumulators to Condition Code Reg.
TBA	Transfer Accumulators
TPA	Transfer Conditions Code Reg. to Accumulator
TST	Test
TSX	Transfer Stack Pointer to Index Register
TXS	Transfer Index Register to Stack Pointer
WAI	Wait for Interrupt

נספח א-2: טבלאות ריכוז הוראות

הוראות דילוג והסתעפות
JUMP AND BRANCH INSTRUCTIONS

הוראה INSTRUCTION	כינוי MNEMONIC	RELATIVE OP	RELATIVE ~	RELATIVE #	INDEX OP	INDEX ~	INDEX #	EXTND OP	EXTND ~	EXTND #	INHER OP	INHER ~	INHER #	בדיקת הסתעפות BRANCH TEST	אוגר מצב COND. CODE REG 5 H	4 I	3 N	2 Z	1 V	0 C
Branch Always	BRA	20	4	2										None	•	•	•	•	•	•
Branch If Carry Clear	BCC	24	4	2										C = 0	•	•	•	•	•	•
Branch If Carry Set	BCS	25	4	2										C = 1	•	•	•	•	•	•
Branch If = Zero	BEQ	27	4	2										Z = 1	•	•	•	•	•	•
Branch If ≥ Zero	BGE	2C	4	2										N ⊕ V = 0	•	•	•	•	•	•
Branch If > Zero	BGT	2E	4	2										Z + (N ⊕ V) = 0	•	•	•	•	•	•
Branch If Higher	BHI	22	4	2										C + Z = 0	•	•	•	•	•	•
Branch If ≤ Zero	BLE	2F	4	2										Z + (N ⊕ V) = 1	•	•	•	•	•	•
Branch If Lower Or Same	BLS	23	4	2										C + Z = 1	•	•	•	•	•	•
Branch If < Zero	BLT	2D	4	2										N ⊕ V = 1	•	•	•	•	•	•
Branch If Minus	BMI	2B	4	2										N = 1	•	•	•	•	•	•
Branch If Not Equal Zero	BNE	26	4	2										Z = 0	•	•	•	•	•	•
Branch If Overflow Clear	BVC	28	4	2										V = 0	•	•	•	•	•	•
Branch If Overflow Set	BVS	29	4	2										V = 1	•	•	•	•	•	•
Branch If Plus	BPL	2A	4	2										N = 0	•	•	•	•	•	•
Branch To Subroutine	BSR	8D	8	2										See Special Operations	•	•	•	•	•	•
Jump	JMP				6E	4	2	7E	3	3				See Special Operations	•	•	•	•	•	•
Jump To Subroutine	JSR				AD	8	2	BD	9	3				See Special Operations	•	•	•	•	•	•
No Operation	NOP										01	2	1	Advances Prog. Cntr. Only	•	•	•	•	•	•
Return From Interrupt	RTI										3B	10	1		(10)					
Return From Subroutine	RTS										39	5	1	See special Operations	•	•	•	•	•	•
Software Interrupt	SWI										3F	12	1	See special Operations	•	S	•	•	•	•
Wait for Interrupt	WAI										3E	9	1	See special Operations	•	(11)	•	•	•	•

(*) המקרא בסוף קבוצת הטבלאות

הוראות אוגר מורה ומחסנית

INDEX REGISTER AND STACK MANIPULATION INSTRUCTIONS

הוראה INSTRUCTION	כינוי MNEMONIC	IMMED OP	IMMED ~	IMMED #	DIRECT OP	DIRECT ~	DIRECT #	INDEX OP	INDEX ~	INDEX #	EXTND OP	EXTND ~	EXTND #	INHER OP	INHER ~	INHER #	מהות הפעולה BOOLEAN/ARITHMETIC OPERATION	אוגר מצב COND. CODE REG 5 H	4 I	3 N	2 Z	1 V	0 C
Compare Index Reg	CPX	8C	3	3	9C	4	2	AC	6	2	BC	5	3				$(X_H/X_L) - (M/M+1)$	•	•	⑦	↕	⑧	•
Decrement Index Reg	DEX													09	4	1	$X - 1 \to X$	•	•	•	↕	•	•
Decrement Stack Pntr	DES													34	4	1	$SP - 1 \to SP$	•	•	•	•	•	•
Increment Index Reg	INX													08	4	1	$X + 1 \to X$	•	•	•	↕	•	•
Increment Stack Pntr	INS													31	4	1	$SP + 1 \to SP$	•	•	•	•	•	•
Load Index Reg	LDX	CE	3	3	DE	4	2	EE	6	2	FE	5	3				$M \to X_H, (M+1) \to X_L$	•	•	⑨	↕	R	•
Load Stack Pntr	LDS	8E	3	3	9E	4	2	AE	6	2	BE	5	3				$M \to SP_H, (M+1) \to SP_L$	•	•	⑨	↕	R	•
Store Index Reg	STX				DF	5	2	EF	7	2	FF	6	3				$X_H \to M, X_L \to (M+1)$	•	•	⑨	↕	R	•
Store Stack Pntr	STS				9F	5	2	AF	7	2	BF	6	3				$SP_H \to M, SP_L \to (M+1)$	•	•	⑨	↕	R	•
Indx Reg → Stack Pntr	TXS													35	4	1	$X - 1 \to SP$	•	•	•	•	•	•
Stack Pntr → Indx Reg	TSX													30	4	1	$SP + 1 \to X$	•	•	•	•	•	•

הוראות אוגר מורה

CONDITION CODE REGISTER MANIPULATION INSTRUCTIONS

הוראה INSTRUCTION	כינוי MNEMONIC	INHER OP	INHER ~	INHER #	מהות הפעולה BOOLEAN OPERATION	5 H	4 I	3 N	2 Z	1 V	0 C
Clear Carry	CLC	0C	2	1	$0 \to C$	•	•	•	•	•	R
Clear Interrupt Mask	CLI	0E	2	1	$0 \to I$	•	R	•	•	•	•
Clear Overflow	CLV	0A	2	1	$0 \to V$	•	•	•	•	R	•
Set Carry	SEC	0D	2	1	$1 \to C$	•	•	•	•	•	S
Set Interrupt Mask	SEI	0F	2	1	$1 \to I$	•	S	•	•	•	•
Set Overflow	SEV	0B	2	1	$1 \to V$	•	•	•	•	S	•
Acmltr A → CCR	TAP	06	2	1	$A \to CCR$	⑫					
CCR → Acmltr A	TPA	07	2	1	$CCR \to A$	•	•	•	•	•	•

CONDITION CODE REGISTER NOTES:

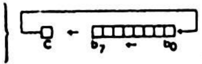
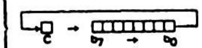
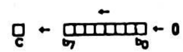
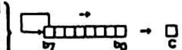
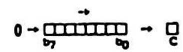
(Bit set if test is true and cleared otherwise)

① (Bit V) Test: Result = 10000000?
② (Bit C) Test: Result ≠ 00000000?
③ (Bit C) Test: Decimal value of most significant BCD Character greater than nine? (Not cleared if previously set.)
④ (Bit V) Test: Operand = 10000000 prior to execution?
⑤ (Bit V) Test: Operand = 01111111 prior to execution?
⑥ (Bit V) Test: Set equal to result of N ⊕ C after shift has occurred.
⑦ (Bit N) Test: Sign bit of most significant (MS) byte of result = 1?
⑧ (Bit V) Test: 2's complement overflow from subtraction of MS bytes?
⑨ (Bit N) Test: Result less than zero? (Bit 15 = 1)
⑩ (All) Load Condition Code Register from Stack. (See Special Operations)
⑪ (Bit I) Set when interrupt occurs. If previously set, a Non-Maskable Interrupt is required to exit the wait state.
⑫ (ALL) Set according to the contents of Accumulator A.

הוראות הצוברים והזכרון

ACCUMULATOR AND MEMORY INSTRUCTIONS

הוראה INSTRUCTION	כינוי MNEMONIC	שיטות מיעון ADDRESSING MODES IMMED			DIRECT			INDEX			EXTND			INHER			מהות הפעולה BOOLEAN/ARITHMETIC OPERATION (All register labels refer to contents)	אוגר מצב COND. CODE REG. 5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Add	ADDA	8B	2	2	9B	3	2	AB	5	2	BB	4	3				A + M → A	↕	●	↕	↕	↕	↕
	ADDB	CB	2	2	DB	3	2	EB	5	2	FB	4	3				B + M → B	↕	●	↕	↕	↕	↕
Add Acmltrs	ABA													1B	2	1	A + B → A	↕	●	↕	↕	↕	↕
Add with Carry	ADCA	89	2	2	99	3	2	A9	5	2	B9	4	3				A + M + C → A	↕	●	↕	↕	↕	↕
	ADCB	C9	2	2	D9	3	2	E9	5	2	F9	4	3				B + M + C → B	↕	●	↕	↕	↕	↕
And	ANDA	84	2	2	94	3	2	A4	5	2	B4	4	3				A · M → A	●	●	↕	↕	R	●
	ANDB	C4	2	2	D4	3	2	E4	5	2	F4	4	3				B · M → B	●	●	↕	↕	R	●
Bit Test	BITA	85	2	2	95	3	2	A5	5	2	B5	4	3				A · M	●	●	↕	↕	R	●
	BITB	C5	2	2	D5	3	2	E5	5	2	F5	4	3				B · M	●	●	↕	↕	R	●
Clear	CLR							6F	7	2	7F	6	3				00 → M	●	●	R	S	R	R
	CLRA													4F	2	1	00 → A	●	●	R	S	R	R
	CLRB													5F	2	1	00 → B	●	●	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	5	2	B1	4	3				A − M	●	●	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	5	2	F1	4	3				B − M	●	●	↕	↕	↕	↕
Compare Acmltrs	CBA													11	2	1	A − B	●	●	↕	↕	↕	↕
Complement, 1's	COM							63	7	2	73	6	3				$\bar{M}$ → M	●	●	↕	↕	R	S
	COMA													43	2	1	$\bar{A}$ → A	●	●	↕	↕	R	S
	COMB													53	2	1	$\bar{B}$ → B	●	●	↕	↕	R	S
Complement, 2's (Negate)	NEG							60	7	2	70	6	3				00 − M → M	●	●	↕	↕	①	②
	NEGA													40	2	1	00 − A → A	●	●	↕	↕	①	②
	NEGB													50	2	1	00 − B → B	●	●	↕	↕	①	②
Decimal Adjust, A	DAA													19	2	1	Converts Binary Add. of BCD Characters into BCD Format	●	●	↕	↕	↕	③
Decrement	DEC							6A	7	2	7A	6	3				M − 1 → M	●	●	↕	↕	④	●
	DECA													4A	2	1	A − 1 → A	●	●	↕	↕	④	●
	DECB													5A	2	1	B − 1 → B	●	●	↕	↕	④	●
Exclusive OR	EORA	88	2	2	98	3	2	A8	5	2	B8	4	3				A ⊕ M → A	●	●	↕	↕	R	●
	EORB	C8	2	2	D8	3	2	E8	5	2	F8	4	3				B ⊕ M → B	●	●	↕	↕	R	●
Increment	INC							6C	7	2	7C	6	3				M + 1 → M	●	●	↕	↕	⑤	●
	INCA													4C	2	1	A + 1 → A	●	●	↕	↕	⑤	●
	INCB													5C	2	1	B + 1 → B	●	●	↕	↕	⑤	●
Load Acmltr	LDAA	86	2	2	96	3	2	A6	5	2	B6	4	3				M → A	●	●	↕	↕	R	●
	LDAB	C6	2	2	D6	3	2	E6	5	2	F6	4	3				M → B	●	●	↕	↕	R	●
Or, Inclusive	ORAA	8A	2	2	9A	3	2	AA	5	2	BA	4	3				A + M → A	●	●	↕	↕	R	●
	ORAB	CA	2	2	DA	3	2	EA	5	2	FA	4	3				B + M → B	●	●	↕	↕	R	●

Push Data	PSHA													36	4	1	A → MSP, SP − 1 → SP	•	•	•	•	•	•
	PSHB													37	4	1	B → MSP, SP − 1 → SP	•	•	•	•	•	•
Pull Data	PULA													32	4	1	SP + 1 → SP, MSP → A	•	•	•	•	•	•
	PULB													33	4	1	SP + 1 → SP, MSP → B	•	•	•	•	•	•
Rotate Left	ROL							69	7	2	79	6	3				M 	•	•	↕	↕	(6)	↕
	ROLA													49	2	1	A	•	•	↕	↕	(6)	↕
	ROLB													59	2	1	B	•	•	↕	↕	(6)	↕
Rotate Right	ROR							66	7	2	76	6	3				M 	•	•	↕	↕	(6)	↕
	RORA													46	2	1	A	•	•	↕	↕	(6)	↕
	RORB													56	2	1	B	•	•	↕	↕	(6)	↕
Shift Left, Arithmetic	ASL							68	7	2	78	6	3				M 	•	•	↕	↕	(6)	↕
	ASLA													48	2	1	A	•	•	↕	↕	(6)	↕
	ASLB													58	2	1	B	•	•	↕	↕	(6)	↕
Shift Right, Arithmetic	ASR							67	7	2	77	6	3				M 	•	•	↕	↕	(6)	↕
	ASRA													47	2	1	A	•	•	↕	↕	(6)	↕
	ASRB													57	2	1	B	•	•	↕	↕	(6)	↕
Shift Right, Logic.	LSR							64	7	2	74	6	3				M 	•	•	R	↕	(6)	↕
	LSRA													44	2	1	A	•	•	R	↕	(6)	↕
	LSRB													54	2	1	B	•	•	R	↕	(6)	↕
Store Acmltr.	STAA				97	4	2	A7	6	2	B7	5	3				A → M	•	•	↕	↕	R	•
	STAB				D7	4	2	E7	6	2	F7	5	3				B → M	•	•	↕	↕	R	•
Subtract	SUBA	80	2	2	90	3	2	A0	5	2	B0	4	3				A − M → A	•	•	↕	↕	↕	↕
	SUBB	C0	2	2	D0	3	2	E0	5	2	F0	4	3				B − M → B	•	•	↕	↕	↕	↕
Subract Acmltrs.	SBA													10	2	1	A − B → A	•	•	↕	↕	↕	↕
Subtr. with Carry	SBCA	82	2	2	92	3	2	A2	5	2	B2	4	3				A − M − C → A	•	•	↕	↕	↕	↕
	SBCB	C2	2	2	D2	3	2	E2	5	2	F2	4	3				B − M − C → B	•	•	↕	↕	↕	↕
Transfer Acmltrs	TAB													16	2	1	A → B	•	•	↕	↕	R	•
	TBA													17	2	1	B → A	•	•	↕	↕	R	•
Test, Zero or Minus	TST							6D	7	2	7D	6	3				M − 00	•	•	↕	↕	R	R
	TSTA													4D	2	1	A − 00	•	•	↕	↕	R	R
	TSTB													5D	2	1	B − 00	•	•	↕	↕	R	R

LEGEND:

OP	Operation Code (Hexadecimal);	M_{SP}	Contents of memory location pointed to be Stack Pointer;	00	Byte = Zero;	R	Reset Always
~	Number of MPU Cycles;	+	Boolean Inclusive OR;	H	Half-carry from bit 3;	S	Set Always
#	Number of Program Bytes;	⊕	Boolean Exclusive OR;	I	Interrupt mask	↕	Test and set if true, cleared otherwise
+	Arithmetic Plus;	$\overline{M}$	Complement of M;	N	Negative (sign bit)	•	Not Affected
−	Arithmetic Minus;	→	Transfer Into;	Z	Zero (byte)	CCR	Condition Code Register
•	Boolean AND;	0	Bit = Zero;	V	Overflow, 2's complement	LS	Least Significant
				C	Carry from bit 7	MS	Most Significant

הוראות ה-6800 לפי קוד פעולה הקסדצימלי
HEXADECIMAL VALUES OF MACHINE CODE

01	NOP			4A	DEC	A		8E	LDS		I
06	TAP			4C	INC	A		90	SUB	A	D
07	TPA			4D	TST	A		91	CMP	A	D
08	INX			4F	CLR	A		92	SBC	A	D
09	DEX			50	NEG	B		94	AND	A	D
0A	CLV			53	COM	B		95	BIT	A	D
0B	SEV			54	LSR	B		96	LDA	A	D
0C	CLC			56	ROR	B		97	STA	A	D
0D	SEC			57	ASR	B		98	EOR	A	D
0E	CLI			58	ASL	B		99	ADC	A	D
0F	SEI			59	ROL	B		9A	ORA	A	D
10	SBA			5A	DEC	B		9B	ADD	A	D
11	CBA			5C	INC	B		9C	CPX		
16	TAB			5D	TST	B		9E	LDS		D
17	TBA			5F	CLR	B		9F	STS		D
19	DAA			60	NEG		X	A0	SUB	A	X
1B	ABA			63	COM		X	A1	CMP	A	X
20	BRA		R	64	LSR		X	A2	SBC	A	X
22	BHI		R	66	ROR		X	A4	AND	A	X
23	BLS		R	67	ASR		X	A5	BIT	A	X
24	BCC		R	68	ASL		X	A6	LDA	A	X
25	BCS		R	69	ROL		X	A7	STA	A	X
26	BNE		R	6A	DEC		X	A8	EOR	A	X
27	BEQ		R	6C	INC		X	A9	ADC	A	X
28	BVC		R	6D	TST		X	AA	ORA	A	X
29	BVS		R	6E	JMP		X	AB	ADD	A	X
2A	BPL		R	6F	CLR		X	AC	CPX		X
2B	BMI		R	70	NEG		E	AD	JSR		X
2C	BGE		R	73	COM		E	AE	LDS		X
2D	BLT		R	74	LSR		E	AF	STS		X
2E	BGT		R	76	ROR		E	B0	SUB	A	E
2F	BLE		R	77	ASR		E	B1	CMP	A	E
30	TSX			78	ASL		E	B2	SBC	A	E
31	INS			79	ROL		E	B4	AND	A	E
32	PUL	A		7A	DEC		E	B5	BIT	A	E
33	PUL	B		7C	INC		E	B6	LDA	A	E
34	DES			7D	TST		E	B7	STA	A	E
35	TXS			7E	JMP		E	B8	EOR	A	E
36	PSH	A		7F	CLR		E	B9	ADC	A	E
37	PSH	B		80	SUB	A	I	BA	ORA	A	E
39	RTS			81	CMP	A	I	BB	ADD	A	E
3B	RTI			82	SBC	A	I	BC	CPX		E
3E	WAI			84	AND	A	I	BD	JSR		E
3F	SWI			85	BIT	A	I	BE	LDS		E
40	NEG	A		86	LDA	A	I	BF	STS		E
43	COM	A		88	EOR	A	I	C0	SUB	B	I
44	LSR	A		89	ADC	A	I	C1	CMP	B	I
46	ROR	A		8A	ORA	A	I	C2	SBC	B	I
47	ASR	A		8B	ADD	A	I	C4	AND	B	I
48	ASL	A		8C	CPX	A	I	C5	BIT	B	I
49	ROL	A		8D	BSR		R	C6	LDA	B	I

C8	EOR	B	I
C9	ADC	B	I
CA	ORA	B	I
CB	ADD	B	I
CE	LDX		I
D0	SUB	B	D
D1	CMP	B	D
D2	SBC	B	D
D4	AND	B	D
D5	BIT	B	D
D6	LSA	B	D
D7	STA	B	D
D8	EOR	B	D
D9	ADC	B	D
DA	ORA	B	D
DB	ORA	B	D
DB	ADD	B	D
DE	LDX		D
DF	STX		D
E0	SUB	B	X
E1	CMP	B	X
E2	SBC	B	X
E4	AND	B	X
E5	BIT	B	X
E6	LDA	B	X
E7	STA	B	X
E8	EOR	B	X
E9	ADC	B	X
EA	ORA	B	X
EB	ADD	B	X
EE	LDX		X
EF	STX	B	E
F0	SUB	B	E
F1	CMP	B	E
F2	SBC	B	E
F4	AND	B	E
F5	BIT	B	E
F6	LDA	B	E
F7	S TA	B	E
F8	ADC	B	E
F9	ADC	B	E
FA	ORA	B	E
FB	ADD	B	E
FE	LDX		E
FF	STX		E

מיקרא:

A צובר = A = Accumulator A

B צובר = B = Accumulator B

מיעון מובלע = = Inherent Addressing

מיעון מיידי = I = Immediate Addressing

מיעון מורחב = E = Extended Addressing

מיעון ישיר מורחב = D = Direct Addressing

מיעון באמצעות אוגר מורה = X = Indexed Addressing

מיעון יחסי = R = Relative Addressing

נספח א-4

מספר מחזורי השעון הדרושים לביצוע הוראות

	Immediate	Direct	Extended	Indexed	Inherent	Relative
ABA	*	*	*	*	2	*
ADC	2	3	4	5	*	*
ADD	2	3	4	5	*	*
AND	2	3	4	5	*	*
ASL	*	*	6	7	2	*
ASR	*	*	6	7	2	*
BCC	*	*	*	*	*	4
BCS	*	*	*	*	*	4
BEA	*	*	*	*	*	4
BGE	*	*	*	*	*	4
BGT	*	*	*	*	*	4
BHI	*	*	*	*	*	4
BIT	2	3	4	5	*	*
BLE	*	*	*	*	*	4
BLS	*	*	*	*	*	4
BLT	*	*	*	*	*	4
BMI	*	*	*	*	*	4
BNE	*	*	*	*	*	4
BPL	*	*	*	*	*	4
BRA	*	*	*	*	*	4
BSR	*	*	*	*	*	8
BVC	*	*	*	*	*	4
BVS	*	*	*	*	*	4
CBA	*	*	*	*	2	*
CLC	*	*	*	*	2	*
CLI	*	*	*	*	2	*
CLR	*	*	6	7	2	*
CLV	*	*	*	*	2	*
CMP	2	3	4	5	*	*
COM	*	*	6	7	2	*
CPX	3	4	5	6	*	*
DAA	*	*	*	*	2	*
DEC	*	*	6	7	2	*
DES	*	*	*	*	4	*
DEX	*	*	*	*	4	*
EOR	2	3	4	5	*	*

	Immediate	Direct	Extended	Indexed	Inherent
INC	*	*	6	7	2
INS	*	*	*	*	4
INX	*	*	*	*	4
JMP	*	*	3	4	*
JSR	*	*	9	8	*
LDA	2	3	4	5	*
LDS	3	4	5	6	*
LDX	3	4	5	6	*
LSR	*	*	6	7	2
NEG	*	*	6	7	2
NOP	*	*	*	*	2
ORA	2	3	4	5	*
PSH	*	*	*	*	4
PUL	*	*	*	*	4
ROL	*	*	6	7	2
ROR	*	*	6	7	2
RTI	*	*	*	*	10
RTS	*	*	*	*	5
SBA	*	*	*	*	2
SBC	2	3	4	5	*
SEC	*	*	*	*	2
SEI	*	*	*	*	2
SEV	*	*	*	*	2
STA	*	4	5	6	*
STS	*	5	6	7	*
STX	*	5	6	7	*
SUB	2	3	4	5	*
SWI	*	*	*	*	12•
TAB	*	*	*	*	2
TAP	*	*	*	*	2
TBA	*	*	*	*	2
TPA	*	*	*	*	2
TST	*	*	6	7	2
TSX	*	*	*	*	4
TXS	*	*	*	*	4
WAI	*	*	*	*	9

•במקרה שהוראה SWI מופיעה מיד לאחר ההוראה WAI הרי דרושים 4 מחזורי שעון בלבד (במקום 12).

נספח ב' – חומרת מערכת עם ה-6800

נספח זה כולל את נתוני היצרן, חברת מוטורולה ואינטל

עבור מרכיבי המערכת הבאים:

- המיקרו מעבד
- זכרונות
- שעון המערכת
- מישק תקשורת מקבילית (PIA)
- מישק תקשורת טורית אסינכרונית (ACIA)
- מישק תקשורת טורית סינכרונית (SSDA)
- מישק תקשורת תעשייתי
- מחשב חד-פיסתי
- זכרון 32K (אינטל)

MOTOROLA

המיקרומעבד 6800

MC6800 (1.0 MHz)
MC68A00 (1.5 MHz)
MC68B00 (2.0 MHz)

MOS
(N-CHANNEL, SILICON-GATE, DEPLETION LOAD)

MICROPROCESSOR



S SUFFIX
CERDIP PACKAGE
CASE 734

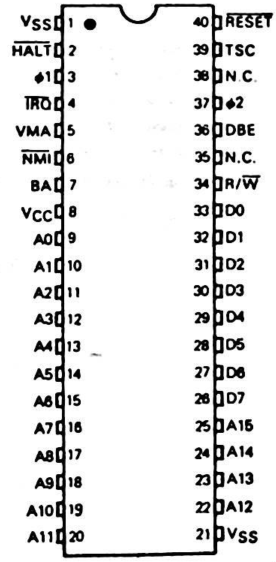


P SUFFIX
PLASTIC PACKAGE
CASE 711



L SUFFIX
CERAMIC PACKAGE
CASE 715

PIN ASSIGNMENT



8-BIT MICROPROCESSING UNIT (MPU)

The MC6800 is a monolithic 8-bit microprocessor forming the central control function for Motorola's M6800 family. Compatible with TTL, the MC6800, as with all M6800 system parts, requires only one +5.0-volt power supply, and no external TTL devices for bus interface.

The MC6800 is capable of addressing 64K bytes of memory with its 16-bit address lines. The 8-bit data bus is bidirectional as well as three-state, making direct memory addressing and multiprocessing applications realizable.

- 8-Bit Parallel Processing
- Bidirectional Data Bus
- 16-Bit Address Bus — 64K Bytes of Addressing
- 72 Instructions — Variable Length
- Seven Addressing Modes — Direct, Relative, Immediate, Indexed, Extended, Implied and Accumulator
- Variable Length Stack
- Vectored Restart
- Maskable Interrupt Vector
- Separate Non-Maskable Interrupt — Internal Registers Saved in Stack
- Six Internal Registers — Two Accumulators, Index Register, Program Counter, Stack Pointer and Condition Code Register
- Direct Memory Addressing (DMA) and Multiple Processor Capability
- Simplified Clocking Characteristics
- Clock Rates as High as 2.0 MHz
- Simple Bus Interface Without TTL
- Halt and Single Instruction Execution Capability

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range MC6800, MC68A00, MC68B00 MC6800C, MC68A00C	T_A	T_L to T_H 0 to +70 −40 to +85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

THERMAL RESISTANCE

Rating	Symbol	Value	Unit
Plastic Package Cerdip Package Ceramic Package	θ_{JA}	100 60 50	°C/W

This device contains circuitry to protect the inputs against damage due to high static voltages or electrical fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage (e.g., either V_{SS} or V_{CC}).

MOS STATIC RAMs (+5 Volts) יחידות זכרון שונות

Organization	Part Number	Access Time (ns max)	No. of Pins
128 × 8	MCM6810	450	24
128 × 8	MCM68A10	360	24
128 × 8	MCM68B10	250	24
1024 × 4	MCM2114P20	200	18
1024 × 4	MCM2114P25	250	18
1024 × 4	MCM2114P30	300	18
1024 × 4	MCM2114P45	450	18
1024 × 4	MCM21L14P20	200	18
1024 × 4	MCM21L14P25	250	18
1024 × 4	MCM21L14P30	300	18
1024 × 4	MCM21L14P45	340	18
1024 × 1	MCM2115AC45[2]	45	16
1024 × 1	MCM2115AC55[2]	55	16
1024 × 1	MCM2115AC70[2]	70	16
1024 × 1	MCM21L15AC45[2]	45	16
1024 × 1	MCM21L15AC70[2]	70	16
1024 × 1	MCM2125AC45	45	16
1024 × 1	MCM2125AC55	55	16
1024 × 1	MCM2125AC70	70	16
1024 × 1	MCM21L25AC45	45	16
1024 × 1	MCM21L25AC70	70	16
4096 × 1	MCM6641P20	200	18
4096 × 1	MCM6641P25	250	18
4096 × 1	MCM6641P30	300	18

BINARY ROMs (+5 Volts)

Organization	Part Number	Access Time (ns max)	No. of Pins
1024 × 8	MCM68A308P	350	24
1024 × 8	MCM68A308P7[4]	350	24
1024 × 8	MCM68B308P	250	24
2048 × 8	MCM68A316EP	350	24
2048 × 8	MCM68A316EP91[4]	350	24
4096 × 8	MCM68A332P	350	24
4096 × 8	MCM68A332P2[4]	350	24
8192 × 8	MCM68A364P	350	24
8192 × 8	MCM68A364P3[4]	350	24
8192 × 8	MCM68B364P	250	24
8192 × 8	MCM68365P25	250	24
8192 × 8	MCM68365P35	350	24
8192 × 8	MCM68366P25	250	24
8192 × 8	MCM68366P35	350	24

MOTOROLA

מישק תקשורת מקבילית
(6821 הוא מודל חדיש של 6820)

MC6821
(1.0 MHz)
MC68A21
(1.5 MHz)
MC68B21
(2.0 MHz)

PERIPHERAL INTERFACE ADAPTER (PIA)

The MC6821 Peripheral Interface Adapter provides the universal means of interfacing peripheral equipment to the M6800 family of microprocessors. This device is capable of interfacing the MPU to peripherals through two 8-bit bidirectional peripheral data buses and four control lines. No external logic is required for interfacing to most peripheral devices.

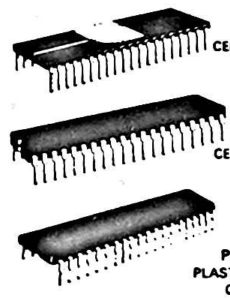
The functional configuration of the PIA is programmed by the MPU during system initialization. Each of the peripheral data lines can be programmed to act as an input or output, and each of the four control/interrupt lines may be programmed for one of several control modes. This allows a high degree of flexibility in the overall operation of the interface.

- 8-Bit Bidirectional Data Bus for Communication with the MPU
- Two Bidirectional 8-Bit Buses for Interface to Peripherals
- Two Programmable Control Registers
- Two Programmable Data Direction Registers
- Four Individually-Controlled Interrupt Input Lines; Two Usable as Peripheral Control Outputs
- Handshake Control Logic for Input and Output Peripheral Operation
- High-Impedance Three-State and Direct Transistor Drive Peripheral Lines
- Program Controlled Interrupt and Interrupt Disable Capability
- CMOS Drive Capability on Side A Peripheral Lines
- Two TTL Drive Capability on All A and B Side Buffers
- TTL-Compatible
- Static Operation

MOS

(N-CHANNEL, SILICON-GATE, DEPLETION LOAD)

PERIPHERAL INTERFACE ADAPTER



L SUFFIX
CERAMIC PACKAGE
CASE 715

S SUFFIX
CERDIP PACKAGE
CASE 734

P SUFFIX
PLASTIC PACKAGE
CASE 711

MAXIMUM RATINGS

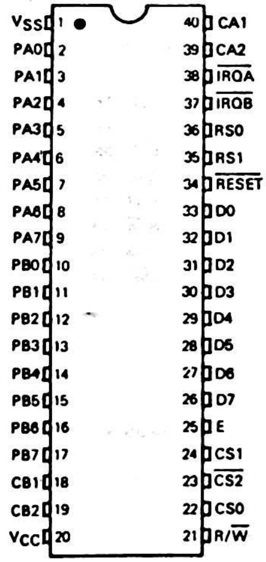
Characteristics	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range MC6821, MC68A21, MC68B21 MC6821C, MC68A21C, MC68B21C	T_A	T_L to T_H 0 to 70 −40 to +85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

THERMAL CHARACTERISTICS

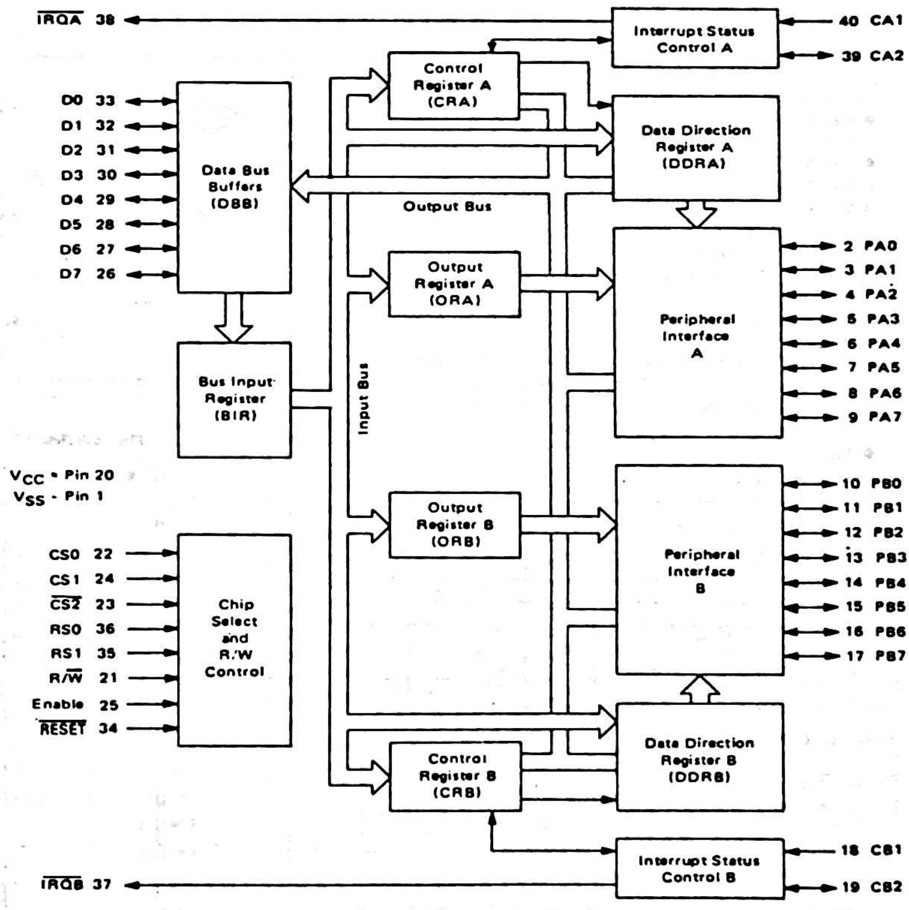
Characteristic	Symbol	Value	Unit
Thermal Resistance Ceramic Plastic Cerdip	θ_{JA}	 50 100 60	°C/W

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage (i.e., either V_{SS} or V_{CC}).

PIN ASSIGNMENT



תרשים מלבנים של מישק תקשורת מקבילית



MOTOROLA

MC6850
(1.0 MHz)
MC68A50
(1.5 MHz)
MC68B50
(2.0 MHz)

מישק תקשורת טורית אסינכרונית

ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER (ACIA)

The MC6850 Asynchronous Communications Interface Adapter provides the data formatting and control to interface serial asynchronous data communications information to bus organized systems such as the MC6800 Microprocessing Unit.

The bus interface of the MC6850 includes select, enable, read/write, interrupt and bus interface logic to allow data transfer over an 8-bit bidirectional data bus. The parallel data of the bus system is serially transmitted and received by the asynchronous data interface, with proper formatting and error checking. The functional configuration of the ACIA is programmed via the data bus during system initialization. A programmable Control Register provides variable word lengths, clock division ratios, transmit control, receive control, and interrupt control. For peripheral or modem operation, three control lines are provided. These lines allow the ACIA to interface directly with the MC6860L 0-600 bps digital modem.

- 8- and 9-Bit Transmission
- Optional Even and Odd Parity
- Parity, Overrun and Framing Error Checking
- Programmable Control Register
- Optional ÷1, ÷16, and ÷64 Clock Modes
- Up to 1.0 Mbps Transmission
- False Start Bit Detection
- Peripheral/Modem Control Functions
- Double Buffered
- One- or Two-Stop Bit Operation

MOS
(N-CHANNEL, SILICON-GATE)

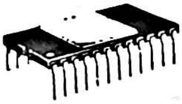
ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER



S SUFFIX
CERDIP PACKAGE
CASE 623

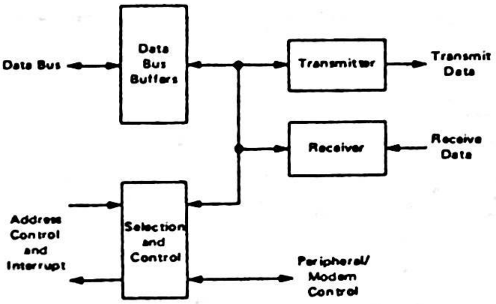


P SUFFIX
PLASTIC PACKAGE
CASE 709

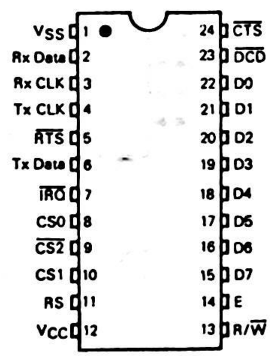


L SUFFIX
CERAMIC PACKAGE
CASE 716

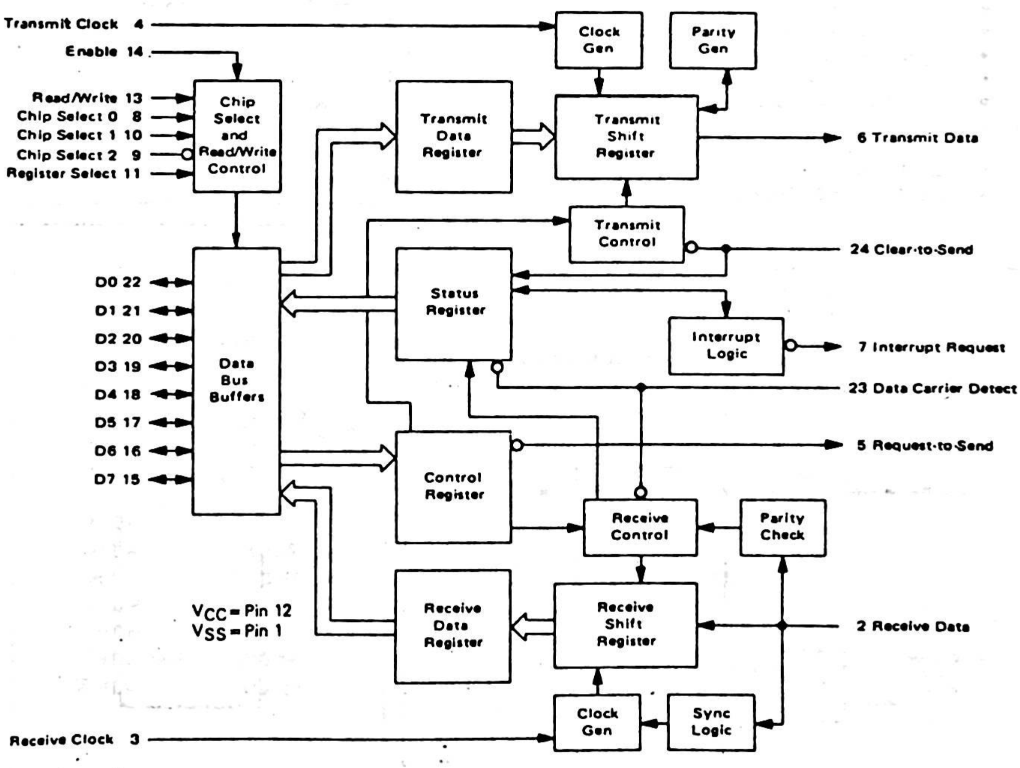
MC6850 ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER BLOCK DIAGRAM



PIN ASSIGNMENT



תרשים מלבנים של מישק תקשורת טורית אסינכרונית



MOTOROLA

מישק תקשורת טורית סנכרונית

MC6852
(1.0 MHz)
MC68A52
(1.5 MHz)
MC68B52
(2.0 MHz)

SYNCHRONOUS SERIAL DATA ADAPTER (SSDA)

The MC6852 Synchronous Serial Data Adapter provides a bidirectional serial interface for synchronous data information interchange. It contains interface logic for simultaneously transmitting and receiving standard synchronous communications characters in bus organized systems such as the M6800 Microprocessor systems.

The bus interface of the MC6852 includes select, enable, read/write, interrupt, and bus interface logic to allow data transfer over an 8-bit bidirectional data bus. The parallel data of the bus system is serially transmitted and received by the synchronous data interface with synchronization, fill character insertion/deletion, and error checking. The functional configuration of the SSDA is programmed via the data bus during system initialization. Programmable control registers provide control for variable word lengths, transmit control, receive control, synchronization control, and interrupt control. Status, timing and control lines provide peripheral or modem control.

Typical applications include floppy disk controllers, cassette or cartridge tape controllers, data communications terminals, and numerical control systems.

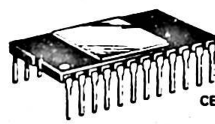
- Programmable Interrupts from Transmitter, Receiver, and Error Detection Logic
- Character Synchronization on One- or Two-Sync Codes
- External Synchronization Available for Parallel-Serial Operation
- Programmable Sync Code Register
- Up to 1.5 MHz Transmission
- Peripheral/Modem Control Functions
- Three Bytes of FIFO Buffering on Both Transmit and Receive
- 7-, 8-, or 9-Bit Transmission
- Optional Even and Odd Parity
- Parity, Overrun, and Underflow Status

MOS
(N-CHANNEL, SILICON-GATE)

SYNCHRONOUS SERIAL DATA ADAPTER



P SUFFIX
PLASTIC PACKAGE
CASE 709

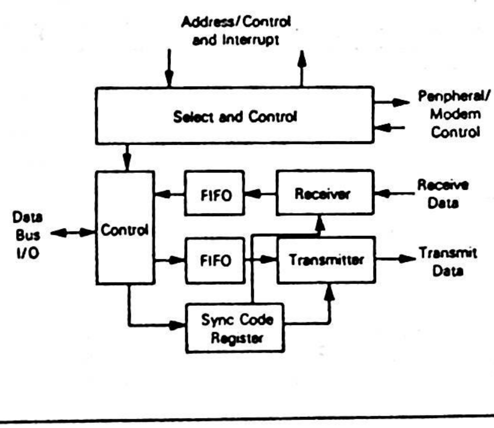


L SUFFIX
CERAMIC PACKAGE
CASE 716

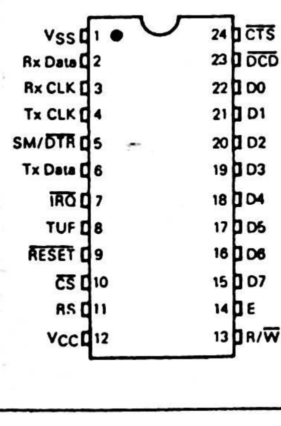


S SUFFIX
CERDIP PACKAGE
CASE 623

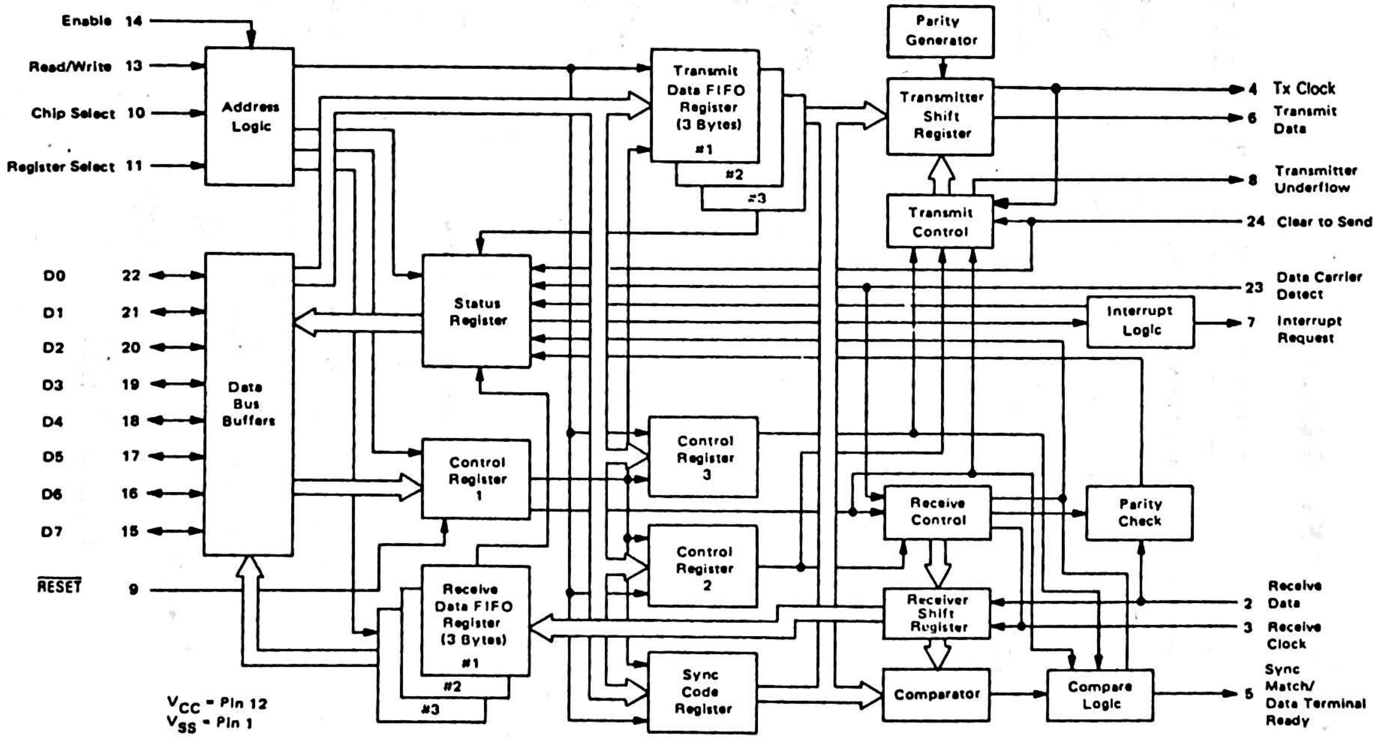
SYNCHRONOUS SERIAL DATA ADAPTER BLOCK DIAGRAM



PIN ASSIGNMENT



תרשים מלבנים של מישק תקשורת טורית סינכרונית



![Motorola logo] **MOTOROLA**

MC6875
MC6875A

שעון המערכת

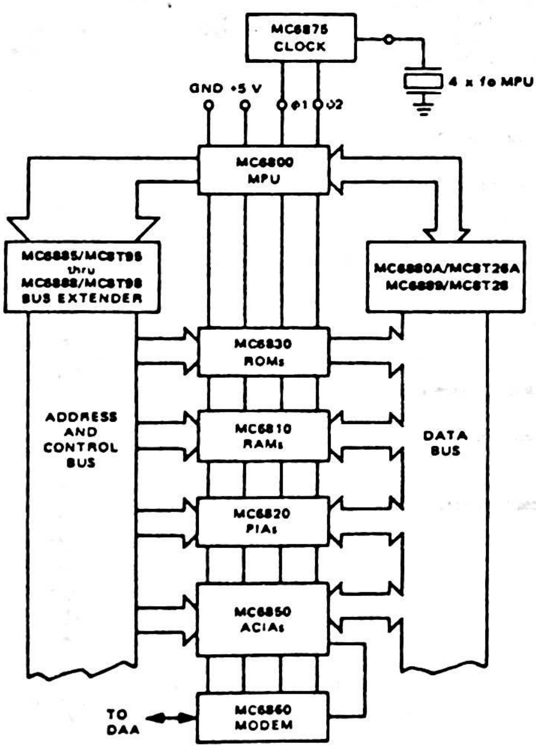
Specifications and Applications Information

M6800 CLOCK GENERATOR

Intended to supply the non-overlapping φ1 and φ2 clock signals required by the microprocessor, this clock generator is compatible with 1.0, 1.5, and 2.0 MHz versions of the MC6800. Both the oscillator and high capacitance driver elements are included along with numerous other logic accessory functions for easy system expansion.

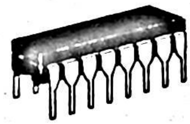
Schottky technology is employed for high speed and PNP-buffered inputs are employed for NMOS compatibility. A single +5 V power supply, and a crystal or RC network for frequency determination are required.

Typical MPU System with Bus Extenders



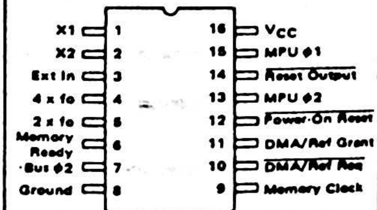
M6800 TWO-PHASE CLOCK GENERATOR/DRIVER

SCHOTTKY MONOLITHIC INTEGRATED CIRCUIT



L SUFFIX
CERAMIC PACKAGE
CASE 620

PIN CONNECTIONS



ORDERING INFORMATION		
Device	Temperature Range	Package
MC6875L	0 to +70°C	Ceramic Dip
MC6875AL	-55 to +125°C	Ceramic Dip

MOTOROLA

מישק תקשורת תעשיתי

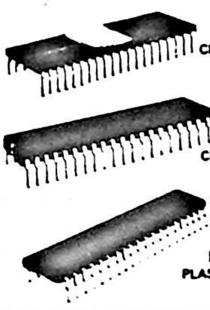
MC6822

Product Preview

MOS

(N-CHANNEL, SILICON-GATE DEPLETION LOAD)

INDUSTRIAL INTERFACE ADAPTER



L SUFFIX
CERAMIC PACKAGE
CASE 715

S SUFFIX
CERDIP PACKAGE
CASE 734

P SUFFIX
PLASTIC PACKAGE
CASE 711

INDUSTRIAL INTERFACE ADAPTER

The MCM6822 Industrial Interface Adapter (IIA) provides the universal means of interfacing industrial peripheral equipment with the MC6800 Microprocessor Unit (MPU). This device is capable of interfacing the MPU with industrial peripherals through two 8-bit bidirectional peripheral parts and four control lines. No external logic is required for interfacing with most peripheral devices. Due to the open-drain design, Peripheral Ports A and B, as well as the peripheral control lines (CA1, CA2, CB1, CB2) can be pulled-up externally to 18.0 V maximum. The recommended operating voltage range for these ports is between 0 and 15 volts. Thus, the IIA can directly interface with 15 V CMOS (no level shifters are required). The IIA is also ideal for industrial applications when increased noise margins are required.

The functional configuration of the IIA is programmed by the MPU during system initialization. Each of the peripheral data lines can be programmed to act as an input or output, and each of the four control/interrupt lines may be programmed for one of several control modes. This allows a high degree of flexibility in the overall operation of the interface.

- 8-Bit Bidirectional Data Bus for Communication with the MPU
- Two Bidirectional 8-Bit Ports for Interface with Peripherals
- Two Programmable Control Registers
- Two Programmable Data Direction Registers
- Four Individually-Controlled Interrupt Lines; Two Usable as Peripheral Control Outputs
- Handshake Control Logic for Input and Output Peripheral Operation
- Open Drain Peripheral Lines Capable of Interfacing with Industrial Equipment
- Program Controlled Interrupt and Interrupt Disable Capability
- CMOS Drive Capability on All Peripheral Lines
- Pin Compatible with MC6821 PIA
- TTL-Compatible Data Bus
- Fully Static Operation

PIN ASSIGNMENT

V_{SS}	1	40	CA1
PA0	2	39	CA2
PA1	3	38	$\overline{IRQA}$
PA2	4	37	$\overline{IRQB}$
PA3	5	36	RS0
PA4	6	35	RS1
PA5	7	34	$\overline{RESET}$
PA6	8	33	D0
PA7	9	32	D1
PB0	10	31	D2
PB1	11	30	D3
PB2	12	29	D4
PB3	13	28	D5
PB4	14	27	D6
PB5	15	26	D7
PB6	16	25	E
PB7	17	24	CS1
CB1	18	23	$\overline{CS2}$
CB2	19	22	CS0
V_{CC}	20	21	$R/\overline{W}$

MOTOROLA

MC68705P3

מחשב חד־פיסתי

Advance Information

8-BIT EPROM MICROCOMPUTER UNIT

The MC68705P3 Microcomputer Unit (MCU) is an EPROM member of the M6805 Family of low-cost single-chip microcomputers. The user programmable EPROM allows program changes and lower volume applications in comparison to the factory mask programmable versions. The EPROM versions also reduce the development costs and turn-around time for prototype evaluation of the mask ROM versions. This 8-bit microcomputer contains a CPU, on-chip CLOCK, EPROM, bootstrap ROM, RAM, I/O, and a TIMER.

Because of these features, the MC68705P3 offers the user an economical means of designing an M6805 Family MCU into his system, either as a prototype evaluation, as a low-volume production run, or a pilot production run.

A comparison table of key features for several members of the M6805 Family is shown on the last page of this data sheet.

HARDWARE FEATURES:

- 8-Bit Architecture
- 112 bytes of RAM
- Memory Mapped I/O
- 1804 Bytes of User EPROM
- Internal 8-Bit Timer with 7-Bit Prescaler
  - Programmable Prescaler
  - Programmable Timer Input Modes
  - External Timer Interrupt
- Vectored Interrupts — External, Timer, and Software
- Zero-Cross Detection on $\overline{INT}$ Input
- 20 TTL/CMOS Compatible Bidirectional I/O Lines (8 Lines are LED Compatible)
- On-Chip Clock Generator
- Master and Power-On Reset
- Complete Development System Support on EXORciser
- Emulates the MC6805P2 and MC6805P4 (Except for V_{SB})
- Bootstrap Program in ROM Simplifies EPROM Programming

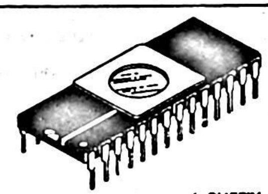
SOFTWARE FEATURES:

- Similar to M6800 Family
- Byte Efficient Instruction Set
- Easy to Program
- True Bit Manipulation
- Bit Test and Branch Instructions
- Versatile Interrupt Handling
- Versatile Index Register
- Powerful Indexed Addressing for Tables
- Full Set of Conditional Branches
- Memory Usable as Registers/Flags
- Single Instruction Memory Examine/Change
- 10 Powerful Addressing Modes
- All Addressing Modes Apply to EPROM, RAM, and I/O

HMOS

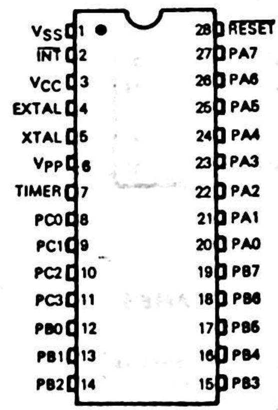
(HIGH-DENSITY, N-CHANNEL DEPLETION LOAD, 5 V EPROM PROCESS)

8-BIT EPROM MICROCOMPUTER



L SUFFIX
CERAMIC PACKAGE
CASE 719

FIGURE 1 — PIN ASSIGNMENTS



intel®

2732
32K (4K x 8) UV ERASABLE PROM

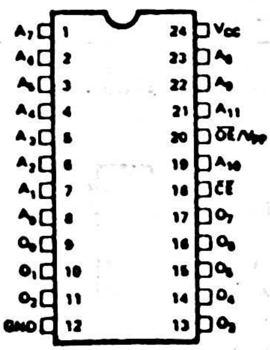
- **Fast Access Time:**
  - — 450 ns Max. 2732
  - — 550 ns Max. 2732-6
- **Single +5V ± 5% Power Supply**
- **Output Enable for MCS-85™ and MCS-86™ Compatibility**
- **Low Power Dissipation:**
  150mA Max. Active Current
  30mA Max. Standby Current
- **Pin Compatible to Intel® 2716 EPROM**
- **Completely Static**
- **Simple Programming Requirements**
  - — Single Location Programming
  - — Programs with One 50ms Pulse
- **Three-State Output for Direct Bus Interface**

The Intel® 2732 is a 32,768-bit ultraviolet erasable and electrically programmable read-only memory (EPROM). The 2732 operates from a single 5-volt power supply, has a standby mode, and features an output enable control. The total programming time for all bits is three and a half minutes. All these features make designing with the 2732 in microcomputer systems faster, easier, and more economical.

An important 2732 feature is the separate output control, Output Enable ($\overline{OE}$) from the Chip Enable control ($\overline{CE}$). The OE control eliminates bus contention in multiple bus microprocessor systems. Intel's Application Note AP-72 describes the microprocessor system implementation of the $\overline{OE}$ and $\overline{CE}$ controls on Intel's 2716 and 2732 EPROMs. AP-72 is available from Intel's Literature Department.

The 2732 has a standby mode which reduces the power dissipation without increasing access time. The maximum active current is 150mA, while the maximum standby current is only 30mA, an 80% savings. The standby mode is achieved by applying a TTL-high signal to the $\overline{CE}$ input.

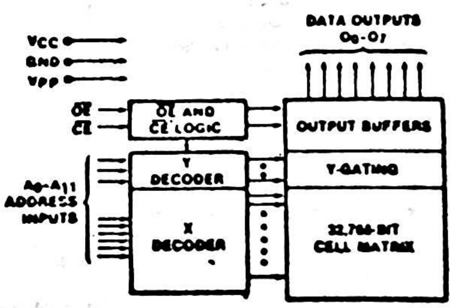
PIN CONFIGURATION



MODE SELECTION

MODE \ PINS	$\overline{CE}$ (18)	$\overline{OE}/V_{PP}$ (20)	V_{CC} (24)	OUTPUTS (9-11,13-17)
Read	V_{IL}	V_{IL}	+5	D_{OUT}
Standby	V_{IH}	Don't Care	+5	High Z
Program	V_{IL}	V_{PP}	+5	D_{IN}
Program Verify	V_{IL}	V_{IL}	+5	D_{OUT}
Program Inhibit	V_{IH}	V_{PP}	+5	High Z

BLOCK DIAGRAM



PIN NAMES

A_0-A_{11}	ADDRESSES
$\overline{CE}$	CHIP ENABLE
$\overline{OE}$	OUTPUT ENABLE
O_0-O_7	OUTPUTS

נספח ג' - טבלאות עזר לתכנות

טבלת מעבר מבסיס ספירה הקסדצימלי לבסיס עשרוני

LSB / MSB	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
8	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
9	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
A	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
B	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
C	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
D	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
E	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
F	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

טבלת מעבר מקוד הקסדצימלי מכוון לבסיס עשרוני

LSB / MSB	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
8	-128	-127	-126	-125	-124	-123	-122	-121	-120	-119	-118	-117	-116	-115	-114	-113
9	-112	-111	-110	-109	-108	-107	-106	-105	-104	-103	-102	-101	-100	-99	-98	-97
A	-96	-95	-94	-93	-92	-91	-90	-89	-88	-87	-86	-85	-84	-83	-82	-81
B	-80	-79	-78	-77	-76	-75	-74	-73	-72	-71	-70	-69	-68	-67	-66	-65
C	-64	-63	-62	-61	-60	-59	-58	-57	-56	-55	-54	-53	-52	-51	-50	-49
D	-48	-47	-46	-45	-44	-43	-42	-41	-40	-39	-38	-37	-36	-35	-34	-33
E	-32	-31	-30	-29	-28	-27	-26	-25	-24	-23	-22	-21	-20	-19	-18	-17
F	-16	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

נספח ד' - מילון מונחים

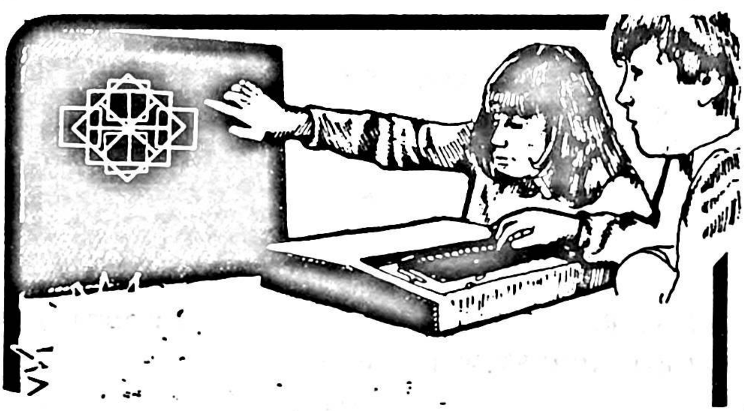
רשימה זו כוללת מונחים באנגלית בנושא שבו דן הספר ותרגומם לעברית כמקובל בבתי-הספר. המספרים מציינים את מספר הסעיף שבו הוגדר המונח.

ACCUMULATOR	2.2	צובר
ADD	6.3	חבר
ADDRESS	1.9	כתובת
ADDRESS BUS; A.B.	2.2	קוי כתובת
ADDRESSING MODES	5.4	שיטות מיעון
AND	6.4	בצע פעולת "גם" לוגי
ARITHMETIC LOGIC UNIT; A.L.U.	2.2	יחידה אריתמטית לוגית
ARITHMETIC SHIFT RIGHT	6.6	הזז ימינה אריתמטית
ASSEMBLER	3.9	תכנית תרגום שפת הסף לשפת המכונה
ASSEMBLY LANGUAGE	3.7	שפת הסף
ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTOR; ACIA	4.6	מישק תקשורת טורית אסינכרונית
BASIC	3.10	בייסיק
BIT	1.4	סיבית
BRANCH ALWAYS; BRA	6.7	הסתעף תמיד
BRANCH IF EQUAL ZERO; BEQ	6.7	הסתעף אם דגל האפס מורם
BRANCH IF MINUS; BMI	6.7	הסתעף אם דגל הכוון מורם
BRANCH TO SUBROUTINE; BSR	6.9	הסתעף לתכנית משנה
BYTE, WORD	1.4	מילה
CARRY/BORROW FLAG; C	5.3.1	דגל הנשא
CENTRAL CLOCK	2.2	שעון מרכזי
CENTRAL PROCESSING UNIT; CPU	1.2	יחידת עיבוד מרכזית, יע"מ
COMPARE; CMP	6.5	השווה
COMPILER	3.10	תוכנית תרגום שפה עילית לשפת מכונה
CLEAR; CLR	6.3	אפס
CLOCK CYCLE	2.11	מחזור השעון
COMPLEMENT; COM	6.4	השלם ל-1
CONDITION CODE REGISTER; CCR	5.3	אוגר מצב
CONTROL BUS; C.B.	2.2	קוי בקרה
CONTROL UNIT; C.U.	2.2	יחידת הבקרה
DATA BUS; D.B.	2.2	קוי נתונים
DEBUGING	4.4	ניפוי (שגיאות)
DIRECT MODE; D	5.4.4	מיעון ישיר
DISK	1.2	תקליט, דיסק
DISKETTE	1.2	תקליטון

OUTPUT	1.2	ציוד פלט
OVERFLOW FLAG; V	5.3.2	דגל ההצפה
PERIPHERAL INTERFACE ADAPTOR; PIA	4.5	מישק תקשורת מקבילית
POWER	2.2	הזנה
PROGRAM COUNTER; P.C.	2.6	מונה תוכנית
PULL; PUL	6.8	משוך
PUSH; PSH	6.8	דחוף
RANDOM ACCESS MEMORY; RAM	4.3	זכרון לקריאה ולכתיבה
READ	2.11	קריאה
READ ONLY MEMORY; ROM	4.4	זכרון לקריאה בלבד
REGISTER	2.2	אוגר
RELATIVE MODE; R	5.4.6	מיעון יחסי
ROTATE RIGHT; ROR	6.6	סובב ימינה
RETURN FROM INTERRUPT; RTI	6.10	חזור מפסיקה
RETURN FROM SUBROUTINE; RTS	6.9	חזור מתוכנית משנה
SET INTERRUPT; SEI	6.5	הרם דגל הפסיקה
SOFTWARE INTERRUPT; SWI	6.10	פסיקת תוכנה
SOURCE PROGRAM	3.8	תוכנית המקור
STACK POINTER; S.P.	2.7	מצביע מחסנית
STORE; ST	6.2	אחסן
SYNCHRONOUS SERIAL DATA ADAPTOR; SSDA	4.7	מישק תקשורת טורית סינכרונית
TEMPORARY REGISTER	2.4	אוגר זמני
TRANSFER; T	6.2	העבר
USER MEMORY	4.3	זכרון למשתמש
USER PROGRAM	3.8	תוכנית המשתמש
USER REGISTERS	5.2	אוגרים בשימוש המתכנת
WAIT FOR INTERRRUPT; WAI	6.10	המתן לפסיקה
WRITE	2.11	כתיבה
ZERO FLAG; Z	5.3.3	דגל האפס

ביבליוגרפיה

1. שלמה אבריצחק, חוברת ללימוד עצמי של מיקרומחשבים.
2. אריה נחום, יסודות המיקרומחשב 6800 ו-8085 לימוד ותרגול, בהוצאת אס.סי.אס. בע"מ.
3. חיים שפירא, מיקרומחשב, בהוצאת אלי, אוקטובר 1981.
4. מיקרופרוססור תוכנה וחומרה, חלקים א', ב', ג' - מט"ח תשמ"ב, 1982.
5. R. Bishop, Basic Microprocessors and the 6800, Hayden Book Company, Inc, 1979.
6. M-6800 Programming Reference Manual, Motorola, Inc. (1976).
7. L.A. Levent, 6800 Assembly Language Programming.
8. A.C. Staugaard, Jr., How to Program and Interface the 6800.
9. Microprocessor Course, Motorola Semiconductor Products, Inc.
10. M-6800 Microcomputer System Design Data, Motorola, Inc.
11. A.A. Cohen, Cumputer Engineering, June 1983.



משחקי ילדים ....

- מהם מיקרומעבד ומיקרומחשב?
- על מה "חושב" המחשב והיכן מסתתרות ההוראות והנתונים?
- באיזו שפה ואיך "מדבר" המחשב?
- מהם הרכיבים והתוכנה של מערכות עם המיקרומעבד 6800?
- איך לתכנת למיקרומחשב עם המיקרו 6800?

מיקרומחשבים וה-6800 – חומרה ותוכנה מציג בפני הקורא והלומד תמונה מקיפה של מערכת החומרה והתוכנה למערכת מיקרומחשב בסיסית כלשהי, ודן במיוחד במערכת שבה מותקן המיקרומעבד 6800. הספר משלב תיאוריה עם הנחיות עבודה, המלצות לתכנות ושאלות לחזרה.

הספר **ערוך** לפי הדרישות של תוכנית הלימודים החדשה של משרד החינוך והתרבות עבור מגמות אלקטרוניקה/מחשבים.

הספר **נועד** למהנדסים ולטכנאים, לסטודנטים ולתלמידים בבתי-ספר תיכוניים-מקצועיים ובבתי-ספר להנדסאים ולמהנדסים, ומתאים ללימוד עצמי על ידי כל המתעניין בנושא.

המחבר
אהרון הראל, יליד 1942, BScTE מטעם אוניברסיטת תל-אביב ובוגר המרכז לחינוך טכנולוגי בחולון. מרכז מגמת אלקטרוניקה/מחשבים בבתי-ספר תיכוניים מקצועיים, מורה למיקרומחשבים ומקצועות אלקטרוניקה שונים, מרצה במכון הטכנולוגי בחולון. בעל נסיון של כ-10 שנים בתחום הבקרה האלקטרונית בתעשיה.

ספרים נוספים בהוצאתנו:

1. תכנון ותכנות יישומי מחשבים, י' עמיהוד
2. הקלט למחשב, ס' מצפי
3. יסודות בניתוח ותכנון מערכות מידע, י' עמיהוד ו-י' קורפל
4. מערכות תקשורת נתונים, א' סקופ
5. ארגון נתונים וניהול קבצים, ר' הייפרמן

כל הספרים באישור משרד החינוך